F29H85x and F29P58x Real-Time Microcontrollers

Technical Reference Manual

Literature Number: SPRUJ79
NOVEMBER 2024

Table of Contents

List of Figures

List of Tables

This page intentionally left blank.

Read This First

About This Manual

This Technical Reference Manual (TRM) details the integration, the environment, the functional description, and the programming models for each peripheral and subsystem in the device.

The TRM should not be considered a substitute for the data sheet, rather a companion guide that can be used alongside the device-specific data sheet to understand the details to program the device. The primary purpose of the TRM is to abstract the programming details of the device from the data sheet. This allows the data sheet to outline the high-level features of the device without unnecessary information about register descriptions or programming models.

Note

Texas Instruments is transitioning to use more inclusive terminology. Some language may be different than what you would expect to see for certain technology areas.

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers can be shown with the suffix h or the prefix 0x. For example, the following number is 40 hexadecimal (decimal 64): 40h or 0x40.
- Registers in this document are shown in figures and described in tables.
  - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties with default reset value below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure can have one of multiple meanings:
    - Not implemented on the device
    - Reserved for future device expansion
    - Reserved for TI testing
    - Reserved configurations of the device that are not supported
  - Writing nondefault values to the Reserved bits could cause unexpected behavior and should be avoided.

Glossary

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

Related Documentation From Texas Instruments

For a complete listing of related documentation and development-support tools for these devices, visit the Texas Instruments website at www.ti.com.

Additionally, the *F29x CPU Reference Guide* must be used in conjunction with this TRM.

Support Resources

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

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

Trademarks

Chapter 1

► C29x SYSTEM RESOURCES

The following chapters describe the C29x Configuration and System Resources.

Technical Reference Manual Overview

The block diagram is shown in Figure 1-1. This Technical Reference Manual is organized into five major sections:

- C29x SYSTEM RESOURCES

  These chapters describe the C29x CPU subsystem, C29x Boot ROM, device configuration, and other system peripherals.

- ANALOG PERIPHERALS

  These chapters describe the general analog subsystem configuration, Analog-to-Digital Converter (ADC), Buffered Digital-to-Analog Converter (DAC), and Comparator Subsystem (CMPSS).

- CONTROL PERIPHERALS

  These chapters describe the Enhanced Capture (eCAP), High-Resolution Capture (HRCAP), Enhanced Pulse-Width Modulator (ePWM) with High-Resolution Pulse-Width Modulator (HRPWM), Enhanced Quadrature Encoder Pulse (eQEP), and Sigma Delta Filter Module (SDFM) peripherals.

- COMMUNICATION PERIPHERALS

  These chapters describe the communication peripherals available to the C29x subsystem such as the EtherCAT, FSI, I2C, PMBUS, UART, LIN, SPI, and SENT.

- SECURITY PERIPHERALS

  This chapter describes the safety peripherals available to the C29x subsystem such as the Hardware Security Module (HSM) and Cryptographic Accelerator.

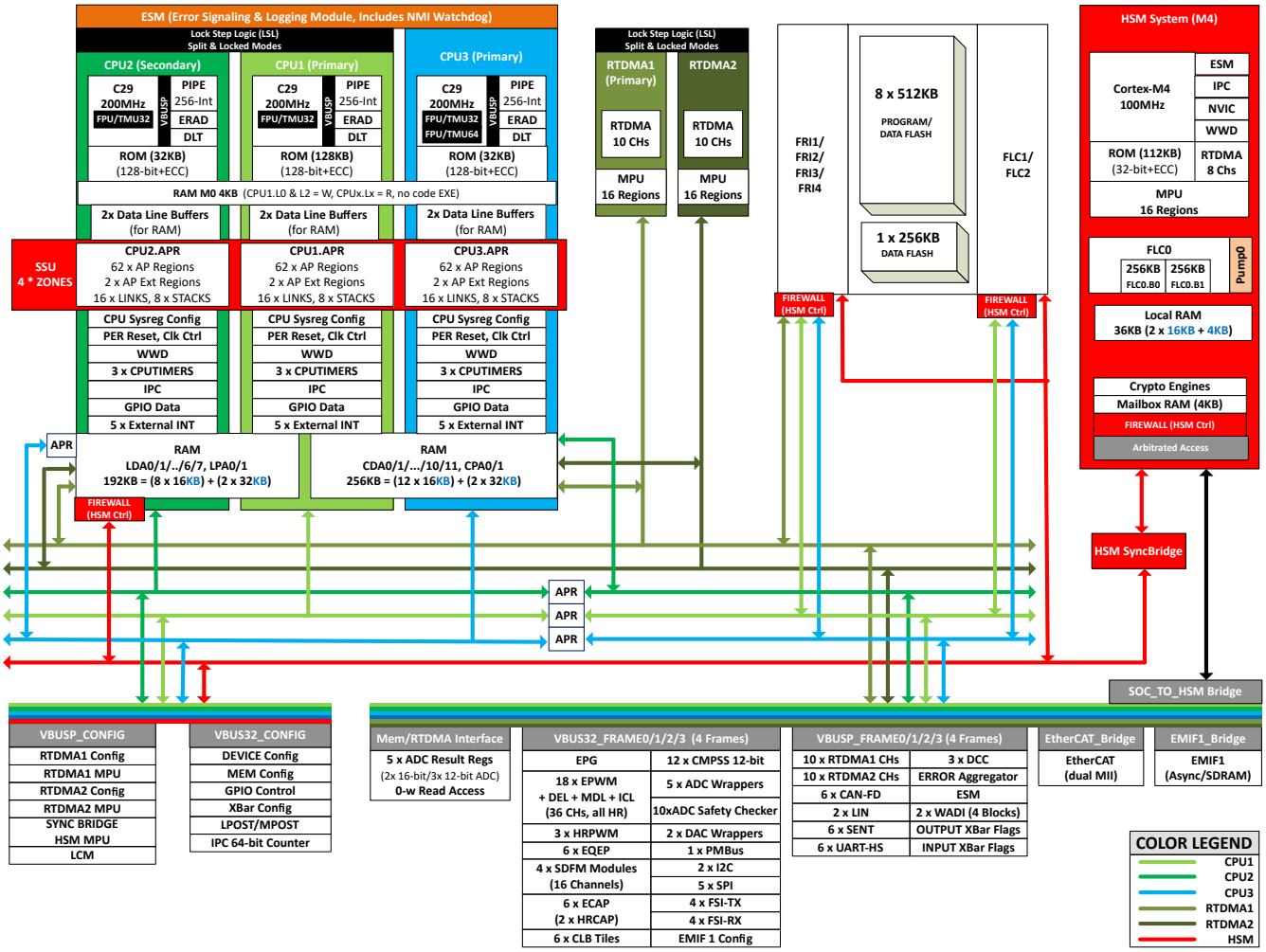


Figure 1-1. Block Diagram

Chapter 2

F29x Processor

This chapter contains a short description of the F29x processor.

Further information can be found in the following documents:

- *F29x CPU Reference Guide*
- *Accelerators: Enhancing the Capabilities of the C2000 MCU Family Technical Brief*

2.1 CPU Architecture

The CPU is a VLIW (Very-Long Instruction Word) architecture with a fully protected pipeline. The CPU supports multiple instruction sizes (16/32/48 bits). The CPU also supports variable instruction packet size, with each packet able to contain up to eight instructions that execute in parallel. For example, the CPU architecture can execute up to eight 16-bit instructions in parallel. This is enabled by multiple functional units inside the CPU which can execute concurrently. A total of 64 working registers, broken into three different categories (Ax, Dx, and Mx Register banks) support the parallel operations in the CPU. In addition to the working registers, the CPU contains multiple status registers (DSTS, ESTS, and ISTS) that maintain execution-related and interrupt-context-related information.

Following are the list of CPU major features:

- **Ease of use:**
  - Byte addressable CPU.
  - Linear and unified memory map with 4GB address range.
  - Fully Protected Pipeline: 9 stage pipeline that prevents writes and reads from same location from occurring out of order.
  - Deterministic execution and maximum performance without cached memories.
- **Improved parallelism:**
  - Execute from 1 to 8 instructions in parallel.
  - Execute fixed-point, floating-point, and addressing operations in parallel.
  - Multiple parallel functional units.
  - Specialized operations to minimize discontinuities and accelerate decision making code (for example, if-then-else statements and switch statements).
  - Specialized operations targeting real-time control (for example, trigonometric operations and multiphase vector translation operations).
- **Improved bus throughput:**
  - Capable of fetching up to 128-bit instruction packet every cycle.
  - Capable of performing 8/16/32/64-bit dual reads and single writes per cycle.
  - Improved addressing modes reduce overhead in accessing memory and peripheral resources.
  - Improved pipeline allows for additional 0-wait memory to be accessible to CPU for max performance.
- **Code efficiency:**
  - Supports variable length instruction set (16-bit, 32-bit, and 48-bit instructions).
  - Rich instruction set optimizes the most common operations in smallest instructions.
- **ASIL-D safety capability with code isolation in hardware:**
  - Lock step core capable of independent execution in split-lock mode (acting as a separate core) or lock step execution (for redundancy).
  - Integrated ECC logic
  - Integrated memory management (MPU) and protection mechanisms in hardware to maximize MIPS.
  - Separate code threads are fully isolated and protected (including software stacks).
- **Multizone security in hardware:**
  - Run time content protection and IP protection of code.
  - Individual passwords for each zone to control access.
- **Enhanced debug and trace capabilities:**
  - Specialized data logging and code flow trace instructions.
  - Trace data capable of being logged in on-chip RAM or exported through serial communication peripherals.

Figure 2-1 shows a block diagram.

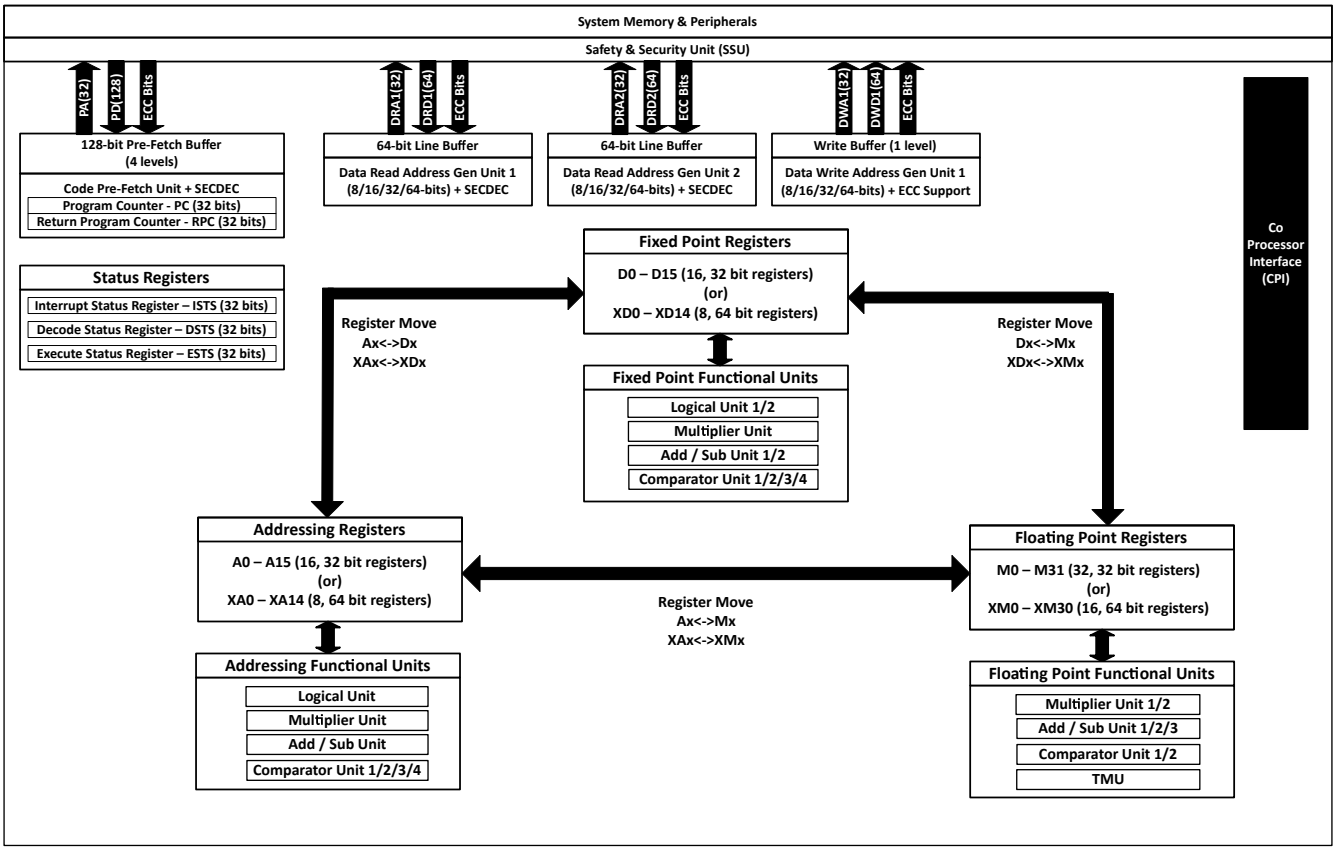


Figure 2-1. CPU Architecture Block Diagram

2.1.1 C29x Related Collateral

Foundational Materials

- C29x Academy
- C29x Academy - Key Literature
- C29x Academy - Key Videos
- C29x Academy - Overview of the C29x CPU

Getting Started Materials

- *Application Software Migration to the C29 CPU*
- *How MCUs Built on Innovative C29 Cores Increase Real-Time Performance in High-Voltage Systems*
- *Real-Time Peripherals Reference Guide*
- *F29x CPU Reference Guide*
- *The new C29 CPU - dominant performance for future real-time applications*
- *User's Guide TMS320F2837x, TMS320F2838x, TMS320F28P65x Migration to F29H85x Application Report*

2.2 Lock and Commit Registers

Lock and Commit mechanism is provided for C29 registers to prevent unintentional or spurious writes to the critical registers.

TI highly recommends that users lock the SECCALL and RTINT stack memory mapped registers using the C29_REGS_LOCK register for fault avoidance from safety aspect and also commit the lock registers using the C29_REGS_COMMIT register for additional security so that registers are not altered during runtime. This can be done at initialization.

2.3 C29x CPU Registers

This section describes the C29x CPU Registers.

2.3.1 C29CPU Base Address Table

Table 2-1. C29CPU Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
C29_RTINT_STACK	C29CPURTINTSTACK_BASE	0x3000_8000	-	YES	YES	YES	-	-	-	YES
C29_SECCALL_STACK	C29CPUSECCALLSTACK_BASE	0x3000_C000	-	YES	YES	YES	-	-	-	YES
C29_SECURE_REGS	C29CPUSECURE_BASE	0x3000_D000	-	YES	YES	YES	-	-	-	YES
C29_DIAG_REGS	C29CPUDIAG_BASE	0x3000_E000	-	YES	YES	YES	-	-	-	YES
C29_SELFTEST_REGS	C29CPUSELFTEST_BASE	0x3000_F000	-	YES	YES	YES	-	-	-	YES

2.3.2 C29_RTINT_STACK Registers

Table 2-2 lists the memory-mapped registers for the C29_RTINT_STACK registers. All register offset addresses not listed in Table 2-2 should be considered as reserved locations and the register contents should not be modified.

Table 2-2. C29_RTINT_STACK Registers

Offset	Acronym	Register Name	Protection
0h + formula	RTINT_STACK_DATA0_j	RTINT stack Data0	
4h + formula	RTINT_STACK_DATA1_j	RTINT stack Data1	
8h + formula	RTINT_STACK_DATA2_j	RTINT stack Data2	
Ch + formula	RTINT_STACK_DATA3_j	RTINT stack Data3	
10h + formula	RTINT_STACK_DATA4_j	RTINT stack Data4	
14h + formula	RTINT_STACK_DATA5_j	RTINT stack Data5	
18h + formula	RTINT_STACK_DATA6_j	RTINT stack Data6	
1Ch + formula	RTINT_STACK_DATA7_j	RTINT stack Data7	
20h + formula	RTINT_STACK_DATA8_j	RTINT stack Data8	
24h + formula	RTINT_STACK_ECC0_j	RTINT stack ECC0	
28h + formula	RTINT_STACK_ECC1_j	RTINT stack ECC1	
2Ch + formula	RTINT_STACK_ECC2_j	RTINT stack ECC2	
30h + formula	RTINT_STACK_ECC3_j	RTINT stack ECC3	

Complex bit access types are encoded to fit into small table cells. Table 2-3 shows the codes that are used for access types in this section.

Table 2-3. C29_RTINT_STACK Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-*n*		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

2.3.2.1 RTINT_STACK_DATA0_j Register (Offset = 0h + formula) [Reset = 00000000h]

RTINT_STACK_DATA0_j is shown in Figure 2-2 and described in Table 2-4.

Return to the Summary Table.

RTINT stack Data0

Offset = 0h + (j * 40h); where j = 0h to 7Fh

Figure 2-2. RTINT_STACK_DATA0_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTINT_STACK_DATA0																															
R/W-0h																															

Table 2-4. RTINT_STACK_DATA0_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTINT_STACK_DATA0	R/W	0h	RTINT stack Bank 0 Data [31:0] Reset type: SYSRSn

2.3.2.2 RTINT_STACK_DATA1_j Register (Offset = 4h + formula) [Reset = 00000000h]

RTINT_STACK_DATA1_j is shown in Figure 2-3 and described in Table 2-5.

Return to the Summary Table.

RTINT stack Data1

Offset = 4h + (j * 40h); where j = 0h to 7Fh

Figure 2-3. RTINT_STACK_DATA1_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTINT_STACK_DATA1																															
R/W-0h																															

Table 2-5. RTINT_STACK_DATA1_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTINT_STACK_DATA1	R/W	0h	RTINT stack Bank 0 Data [63:32] Reset type: SYSRSn

2.3.2.3 RTINT_STACK_DATA2_j Register (Offset = 8h + formula) [Reset = 00000000h]

RTINT_STACK_DATA2_j is shown in Figure 2-4 and described in Table 2-6.

Return to the Summary Table.

RTINT stack Data2

Offset = 8h + (j * 40h); where j = 0h to 7Fh

Figure 2-4. RTINT_STACK_DATA2_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTINT_STACK_DATA2																															
R/W-0h																															

Table 2-6. RTINT_STACK_DATA2_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTINT_STACK_DATA2	R/W	0h	RTINT stack Bank 0 Data [71:64], RTINT stack Bank 1 Data [23:0] Reset type: SYSRSn

2.3.2.4 RTINT_STACK_DATA3_j Register (Offset = Ch + formula) [Reset = 00000000h]

RTINT_STACK_DATA3_j is shown in Figure 2-5 and described in Table 2-7.

Return to the Summary Table.

RTINT stack Data3

Offset = Ch + (j * 40h); where j = 0h to 7Fh

Figure 2-5. RTINT_STACK_DATA3_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTINT_STACK_DATA3																															
R/W-0h																															

Table 2-7. RTINT_STACK_DATA3_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTINT_STACK_DATA3	R/W	0h	RTINT stack Bank 1 Data [55:24] Reset type: SYSRSn

2.3.2.5 RTINT_STACK_DATA4_j Register (Offset = 10h + formula) [Reset = 00000000h]

RTINT_STACK_DATA4_j is shown in Figure 2-6 and described in Table 2-8.

Return to the Summary Table.

RTINT stack Data4

Offset = 10h + (j * 40h); where j = 0h to 7Fh

Figure 2-6. RTINT_STACK_DATA4_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTINT_STACK_DATA4																															
R/W-0h																															

Table 2-8. RTINT_STACK_DATA4_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTINT_STACK_DATA4	R/W	0h	RTINT stack Bank 1 Data [71:56], RTINT stack Bank 2 Data [15:0] Reset type: SYSRSn

2.3.2.6 RTINT_STACK_DATA5_j Register (Offset = 14h + formula) [Reset = 00000000h]

RTINT_STACK_DATA5_j is shown in Figure 2-7 and described in Table 2-9.

Return to the Summary Table.

RTINT stack Data5

Offset = 14h + (j * 40h); where j = 0h to 7Fh

Figure 2-7. RTINT_STACK_DATA5_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTINT_STACK_DATA5																															
R/W-0h																															

Table 2-9. RTINT_STACK_DATA5_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTINT_STACK_DATA5	R/W	0h	RTINT stack Bank 2 Data [47:16] Reset type: SYSRSn

2.3.2.7 RTINT_STACK_DATA6_j Register (Offset = 18h + formula) [Reset = 00000000h]

RTINT_STACK_DATA6_j is shown in Figure 2-8 and described in Table 2-10.

Return to the Summary Table.

RTINT stack Data6

Offset = 18h + (j * 40h); where j = 0h to 7Fh

Figure 2-8. RTINT_STACK_DATA6_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTINT_STACK_DATA6																															
R/W-0h																															

Table 2-10. RTINT_STACK_DATA6_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTINT_STACK_DATA6	R/W	0h	RTINT stack Bank 2 Data [71:48], RTINT stack Bank 3 Data [7:0] Reset type: SYSRSn

2.3.2.8 RTINT_STACK_DATA7_j Register (Offset = 1Ch + formula) [Reset = 00000000h]

RTINT_STACK_DATA7_j is shown in Figure 2-9 and described in Table 2-11.

Return to the Summary Table.

RTINT stack Data7

Offset = 1Ch + (j * 40h); where j = 0h to 7Fh

Figure 2-9. RTINT_STACK_DATA7_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTINT_STACK_DATA7																															
R/W-0h																															

Table 2-11. RTINT_STACK_DATA7_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTINT_STACK_DATA7	R/W	0h	RTINT stack Bank 3 Data [39:8] Reset type: SYSRSn

2.3.2.9 RTINT_STACK_DATA8_j Register (Offset = 20h + formula) [Reset = 00000000h]

RTINT_STACK_DATA8_j is shown in Figure 2-10 and described in Table 2-12.

Return to the Summary Table.

RTINT stack Data8

Offset = 20h + (j * 40h); where j = 0h to 7Fh

Figure 2-10. RTINT_STACK_DATA8_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTINT_STACK_DATA8																															
R/W-0h																															

Table 2-12. RTINT_STACK_DATA8_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTINT_STACK_DATA8	R/W	0h	RTINT stack Bank 3 Data [71:40] Reset type: SYSRSn

2.3.2.10 RTINT_STACK_ECC0_j Register (Offset = 24h + formula) [Reset = 00000000h]

RTINT_STACK_ECC0_j is shown in Figure 2-11 and described in Table 2-13.

Return to the Summary Table.

RTINT stack ECC0

Offset = 24h + (j * 40h); where j = 0h to 7Fh

Figure 2-11. RTINT_STACK_ECC0_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RTINT_STACK_ECC0							
R-0h								R/W-0h							

Table 2-13. RTINT_STACK_ECC0_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	RTINT_STACK_ECC0	R/W	0h	RTINT stack ECC0 - Bank0 (#) Data[79:72] Reset type: SYSRSn

2.3.2.11 RTINT_STACK_ECC1_j Register (Offset = 28h + formula) [Reset = 00000000h]

RTINT_STACK_ECC1_j is shown in Figure 2-12 and described in Table 2-14.

Return to the Summary Table.

RTINT stack ECC1

Offset = 28h + (j * 40h); where j = 0h to 7Fh

Figure 2-12. RTINT_STACK_ECC1_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RTINT_STACK_ECC1							
R-0h								R/W-0h							

Table 2-14. RTINT_STACK_ECC1_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	RTINT_STACK_ECC1	R/W	0h	RTINT stack ECC1 - Bank1 (#) Data[79:72] Reset type: SYSRSn

2.3.2.12 RTINT_STACK_ECC2_j Register (Offset = 2Ch + formula) [Reset = 00000000h]

RTINT_STACK_ECC2_j is shown in Figure 2-13 and described in Table 2-15.

Return to the Summary Table.

RTINT stack ECC2

Offset = 2Ch + (j * 40h); where j = 0h to 7Fh

Figure 2-13. RTINT_STACK_ECC2_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RTINT_STACK_ECC2							
R-0h								R/W-0h							

Table 2-15. RTINT_STACK_ECC2_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	RTINT_STACK_ECC2	R/W	0h	RTINT stack ECC2 - Bank2 (#) Data[79:72] Reset type: SYSRSn

2.3.2.13 RTINT_STACK_ECC3_j Register (Offset = 30h + formula) [Reset = 00000000h]

RTINT_STACK_ECC3_j is shown in Figure 2-14 and described in Table 2-16.

Return to the Summary Table.

RTINT stack ECC3

Offset = 30h + (j * 40h); where j = 0h to 7Fh

Figure 2-14. RTINT_STACK_ECC3_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RTINT_STACK_ECC3							
R-0h								R/W-0h							

Table 2-16. RTINT_STACK_ECC3_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	RTINT_STACK_ECC3	R/W	0h	RTINT stack ECC3 - Bank3 (#) Data[79:72] Reset type: SYSRSn

2.3.3 C29_SECCALL_STACK Registers

Table 2-17 lists the memory-mapped registers for the C29_SECCALL_STACK registers. All register offset addresses not listed in Table 2-17 should be considered as reserved locations and the register contents should not be modified.

Table 2-17. C29_SECCALL_STACK Registers

Offset	Acronym	Register Name	Protection
0h + formula	SECCALL_STACK_DATA0_j	SECCALL stack Data0	
4h + formula	SECCALL_STACK_DATA1_j	SECCALL stack Data1	
8h + formula	SECCALL_STACK_DATA2_j	SECCALL stack Data2	

Complex bit access types are encoded to fit into small table cells. Table 2-18 shows the codes that are used for access types in this section.

Table 2-18. C29_SECCALL_STACK Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

2.3.3.1 SECCALL_STACK_DATA0_j Register (Offset = 0h + formula) [Reset = 00000000h]

SECCALL_STACK_DATA0_j is shown in Figure 2-15 and described in Table 2-19.

Return to the Summary Table.

SECCALL stack Data0

Offset = 0h + (j * 10h); where j = 0h to 10h

Figure 2-15. SECCALL_STACK_DATA0_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SECCALL_STACK_DATA0																															
R/W-0h																															

Table 2-19. SECCALL_STACK_DATA0_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SECCALL_STACK_DATA0	R/W	0h	Secure call stack Data [31:0] Reset type: SYSRSn

2.3.3.2 SECCALL_STACK_DATA1_j Register (Offset = 4h + formula) [Reset = 00000000h]

SECCALL_STACK_DATA1_j is shown in Figure 2-16 and described in Table 2-20.

Return to the Summary Table.

SECCALL stack Data1

Offset = 4h + (j * 10h); where j = 0h to 10h

Figure 2-16. SECCALL_STACK_DATA1_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SECCALL_STACK_DATA1																															
R/W-0h																															

Table 2-20. SECCALL_STACK_DATA1_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SECCALL_STACK_DATA1	R/W	0h	Secure call stack Data [63:32] Reset type: SYSRSn

2.3.3.3 SECCALL_STACK_DATA2_j Register (Offset = 8h + formula) [Reset = 00000000h]

SECCALL_STACK_DATA2_j is shown in Figure 2-17 and described in Table 2-21.

Return to the Summary Table.

SECCALL stack Data2

Offset = 8h + (j * 10h); where j = 0h to 10h

Figure 2-17. SECCALL_STACK_DATA2_j Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				SECCALL_STACK_DATA2			
R-0h				R/W-0h			

Table 2-21. SECCALL_STACK_DATA2_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	SECCALL_STACK_DATA2	R/W	0h	Secure call stack Data [67:64] Reset type: SYSRSn

2.3.4 C29_SECURE_REGS Registers

Table 2-22 lists the memory-mapped registers for the C29_SECURE_REGS registers. All register offset addresses not listed in Table 2-22 should be considered as reserved locations and the register contents should not be modified.

Table 2-22. C29_SECURE_REGS Registers

Offset	Acronym	Register Name	Protection
0h	SECSP0	Secure Stackpointer 0	
4h	SECSP1	Secure Stackpointer 1	
8h	SECSP2	Secure Stackpointer 2	
Ch	SECSP3	Secure Stackpointer 3	
10h	SECSP4	Secure Stackpointer 4	
14h	SECSP5	Secure Stackpointer 5	
18h	SECSP6	Secure Stackpointer 6	
1Ch	SECSP7	Secure Stackpointer 7	
80h	PSP	Protected call stack pointer	
84h	WARNPSP	Warning level register for protected call stack pointer	
88h	MAXPSP	Maximum levels of protected calls supported by the HW	
8Ch	REVISION	IP revision id register	
90h	C29_REGS_LOCK	Lock register	
94h	C29_REGS_COMMIT	Commit register	

Complex bit access types are encoded to fit into small table cells. Table 2-23 shows the codes that are used for access types in this section.

Table 2-23. C29_SECURE_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

2.3.4.1 SECSP0 Register (Offset = 0h) [Reset = 00000000h]

SECSP0 is shown in Figure 2-18 and described in Table 2-24.

Return to the Summary Table.

Secure Stackpointer 0

Figure 2-18. SECSP0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SP0																															
R/W-0h																															

Table 2-24. SECSP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SP0	R/W	0h	Secure stack pointer 0 Only 32-bit access is permitted to this register, shorter accesses are not a use-case Reset type: SYSRSn

2.3.4.2 SECSP1 Register (Offset = 4h) [Reset = 00000000h]

SECSP1 is shown in Figure 2-19 and described in Table 2-25.

Return to the Summary Table.

Secure Stackpointer 1

Figure 2-19. SECSP1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SP1																															
R/W-0h																															

Table 2-25. SECSP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SP1	R/W	0h	Secure stack pointer 1 Only 32-bit access is permitted to this register, shorter accesses are not a use-case Reset type: SYSRSn

2.3.4.3 SECSP2 Register (Offset = 8h) [Reset = 00000000h]

SECSP2 is shown in Figure 2-20 and described in Table 2-26.

Return to the Summary Table.

Secure Stackpointer 2

Figure 2-20. SECSP2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SP2																															
R/W-0h																															

Table 2-26. SECSP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SP2	R/W	0h	Secure stack pointer 2 Only 32-bit access is permitted to this register, shorter accesses are not a use-case Reset type: SYSRSn

2.3.4.4 SECSP3 Register (Offset = Ch) [Reset = 00000000h]

SECSP3 is shown in Figure 2-21 and described in Table 2-27.

Return to the Summary Table.

Secure Stackpointer 3

Figure 2-21. SECSP3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SP3																															
R/W-0h																															

Table 2-27. SECSP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SP3	R/W	0h	Secure stack pointer 3 Only 32-bit access is permitted to this register, shorter accesses are not a use-case Reset type: SYSRSn

2.3.4.5 SECSP4 Register (Offset = 10h) [Reset = 00000000h]

SECSP4 is shown in Figure 2-22 and described in Table 2-28.

Return to the Summary Table.

Secure Stackpointer 4

Figure 2-22. SECSP4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SP4																															
R/W-0h																															

Table 2-28. SECSP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SP4	R/W	0h	Secure stack pointer 4 Only 32-bit access is permitted to this register, shorter accesses are not a use-case Reset type: SYSRSn

2.3.4.6 SECSP5 Register (Offset = 14h) [Reset = 00000000h]

SECSP5 is shown in Figure 2-23 and described in Table 2-29.

Return to the Summary Table.

Secure Stackpointer 5

Figure 2-23. SECSP5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SP5																															
R/W-0h																															

Table 2-29. SECSP5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SP5	R/W	0h	Secure stack pointer 5 Only 32-bit access is permitted to this register, shorter accesses are not a use-case Reset type: SYSRSn

2.3.4.7 SECSP6 Register (Offset = 18h) [Reset = 00000000h]

SECSP6 is shown in Figure 2-24 and described in Table 2-30.

Return to the Summary Table.

Secure Stackpointer 6

Figure 2-24. SECSP6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SP6																															
R/W-0h																															

Table 2-30. SECSP6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SP6	R/W	0h	Secure stack pointer 6 Only 32-bit access is permitted to this register, shorter accesses are not a use-case Reset type: SYSRSn

2.3.4.8 SECSP7 Register (Offset = 1Ch) [Reset = 00000000h]

SECSP7 is shown in Figure 2-25 and described in Table 2-31.

Return to the Summary Table.

Secure Stackpointer 7

Figure 2-25. SECSP7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SP7																															
R/W-0h																															

Table 2-31. SECSP7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SP7	R/W	0h	Secure stack pointer 7 Only 32-bit access is permitted to this register, shorter accesses are not a use-case Reset type: SYSRSn

2.3.4.9 PSP Register (Offset = 80h) [Reset = 00000000h]

PSP is shown in Figure 2-26 and described in Table 2-32.

Return to the Summary Table.

Protected call stack pointer

Figure 2-26. PSP Register

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

Table 2-32. PSP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11-0	PSP	R	0h	Current value of protected call stack pointer. Increamented in HW on Secure Call and decremeted on Secure Return. It can be initialized by the CPU provided SSU security rules are met. Reset type: SYSRSn

2.3.4.10 WARNPSP Register (Offset = 84h) [Reset = 0000000Ch]

WARNPSP is shown in Figure 2-27 and described in Table 2-33.

Return to the Summary Table.

Warning level register for protected call stack pointer

Figure 2-27. WARNPSP Register

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

Table 2-33. WARNPSP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11-0	WARNPSP	R/W	Ch	Warning level for protected call stack pointer.It can be initialized by the CPU provided SSU security rules are met. External logic may compare PSP with WARNPSP to generate early warning interrupt to avoid protected call stack overflow. Reset type: SYSRSn

2.3.4.11 MAXPSP Register (Offset = 88h) [Reset = 00000010h]

MAXPSP is shown in Figure 2-28 and described in Table 2-34.

Return to the Summary Table.

Maximum levels of protected calls supported by the HW

Figure 2-28. MAXPSP Register

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

Table 2-34. MAXPSP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11-0	MAXPSP	R	10h	Maximum levels of protected calls supported by the HW.It can be initialized by the CPU provided SSU security rules are met. External logic may compare PSP with MAXPSP to generate NMI interrupt to indicate stack over flow. Reset type: SYSRSn

2.3.4.12 REVISION Register (Offset = 8Ch) [Reset = 00009800h]

REVISION is shown in Figure 2-29 and described in Table 2-35.

Return to the Summary Table.

IP revision id register

Figure 2-29. REVISION Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
MAJREV_CPUCONFIG			MAJREV_TMU64_PRESENT	MAJREV_FPU64_PRESENT	MAJREV_Revision		
R-4h			R-1h	R-1h	R-0h		
7	6	5	4	3	2	1	0
MINREV							
R-0h							

Table 2-35. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-13	MAJREV_CPUCONFIG	R	4h	This hardcoded field defines CPU configuration 000 : Low end 100 : Base configuration(Veloce) 111 : Higher end Reset type: SYSRSn
12	MAJREV_TMU64_PRESENT	R	1h	This hardcoded field defines whether TMU64 is present in the IP. Reset type: SYSRSn
11	MAJREV_FPU64_PRESENT	R	1h	This hardcoded field defines whether FPU64 is present in the IP. Reset type: SYSRSn
10-8	MAJREV_Revision	R	0h	This hardcoded field defines the major revision of the IP. Reset type: SYSRSn
7-0	MINREV	R	0h	This hardcoded field defines the minor revision of the IP. Reset type: SYSRSn

2.3.4.13 C29_REGS_LOCK Register (Offset = 90h) [Reset = 00000000h]

C29_REGS_LOCK is shown in Figure 2-30 and described in Table 2-36.

Return to the Summary Table.

C29 registers lock

Figure 2-30. C29_REGS_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		C29_SELFTEST_REGS	C29_DIAG_REGS	C29_SECURE_REGS	C29_SECCALL_STACK	C29_RTINT_STACK	RESERVED
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 2-36. C29_REGS_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5	C29_SELFTEST_REGS	R/W	0h	When set, locks the C29 SELFTEST register region (writes will have no effect on it). This bit can only be modified if C29_REGS_COMMIT.C29_SELFTEST_REGS is cleared. 0 : Unlocked 1 : Locked Reset type: SYSRSn
4	C29_DIAG_REGS	R/W	0h	When set, locks the C29 DIAG register region (writes will have no effect on it). This bit can only be modified if C29_REGS_COMMIT.C29_DIAG_REGS is cleared. 0 : Unlocked 1 : Locked Reset type: SYSRSn
3	C29_SECURE_REGS	R/W	0h	When set, locks the C29 SECURE register region (writes will have no effect on it). This bit can only be modified if C29_REGS_COMMIT.C29_SECURE_REGST is cleared. 0 : Unlocked 1 : Locked Reset type: SYSRSn
2	C29_SECCALL_STACK	R/W	0h	When set, locks the C29 SECCALL stack register region (writes will have no effect on it). This bit can only be modified if C29_REGS_COMMIT.C29_SECCALL_STACK is cleared. 0 : Unlocked 1 : Locked Reset type: SYSRSn
1	C29_RTINT_STACK	R/W	0h	When set, locks the C29 RTINT stack register region (writes will have no effect on it). This bit can only be modified if C29_REGS_COMMIT.C29_RTINT_STACK is cleared. 0 : Unlocked 1 : Locked Reset type: SYSRSn

Table 2-36. C29_REGS_LOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	RESERVED	R	0h	Reserved

2.3.4.14 C29_REGS_COMMIT Register (Offset = 94h) [Reset = 00000000h]

C29_REGS_COMMIT is shown in Figure 2-31 and described in Table 2-37.

Return to the Summary Table.

C29 registers commit

Figure 2-31. C29_REGS_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		C29_SELFTEST_REGS	C29_DIAG_REGS	C29_SECURE_REGS	C29_SECCALL_STACK	C29_RTINT_STACK	RESERVED
R-0h		R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R-0h

Table 2-37. C29_REGS_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5	C29_SELFTEST_REGS	R/W1S	0h	When set, locks C29_REGS_LOCK.C29_SELFTEST_REGS register field (writes will have no effect on it). This bit cannot be cleared, except by reset. 0 : C29_REGS_LOCK.C29_SELFTEST_REGS is modifiable 1 : C29_REGS_LOCK.C29_SELFTEST_REGS is committed permanently Reset type: SYSRSn
4	C29_DIAG_REGS	R/W1S	0h	When set, locks C29_REGS_LOCK.C29_DIAG_REGS register field (writes will have no effect on it). This bit cannot be cleared, except by reset. 0 : C29_REGS_LOCK.C29_DIAG_REGS is modifiable 1 : C29_REGS_LOCK.C29_DIAG_REGS is committed permanently Reset type: SYSRSn
3	C29_SECURE_REGS	R/W1S	0h	When set, locks C29_REGS_LOCK.C29_SECURE_REGS register field (writes will have no effect on it). This bit cannot be cleared, except by reset. 0 : C29_REGS_LOCK.C29_SECURE_REGS is modifiable 1 : C29_REGS_LOCK.C29_SECURE_REGS is committed permanently Reset type: SYSRSn
2	C29_SECCALL_STACK	R/W1S	0h	When set, locks C29_REGS_LOCK.C29_SECCALL_STACK register field (writes will have no effect on it). This bit cannot be cleared, except by reset. 0 : C29_REGS_LOCK.C29_SECCALL_STACK is modifiable 1 : C29_REGS_LOCK.C29_SECCALL_STACK is committed permanently Reset type: SYSRSn

Table 2-37. C29_REGS_COMMIT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	C29_RTINT_STACK	R/W1S	0h	When set, locks C29_REGS_LOCK.C29_RTINT_STACK register field (writes will have no effect on it). This bit cannot be cleared, except by reset. 0 : C29_REGS_LOCK.C29_RTINT_STACK is modifiable 1 : C29_REGS_LOCK.C29_RTINT_STACK is committed permanently Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

2.3.5 C29_DIAG_REGS Registers

Table 2-38 lists the memory-mapped registers for the C29_DIAG_REGS registers. All register offset addresses not listed in Table 2-38 should be considered as reserved locations and the register contents should not be modified.

Table 2-38. C29_DIAG_REGS Registers

Offset	Acronym	Register Name	Protection
0h	FLTEMU_CONFIG	Fault emulation configuration registerr	KEY:KEY=0xa5
4h	FLTEMU_ACCGRPSEL	Fault emulation access information group selection register	
8h	FLTEMU_BITSEL	Fault emulation bitsel	
Ch	FLTEMU_ADDR	Fault emulation access address register	
10h	TMU_ROM_PAR_FORCE	Force TMU ROM Parity error	KEY:KEY=0xa5

Complex bit access types are encoded to fit into small table cells. Table 2-39 shows the codes that are used for access types in this section.

Table 2-39. C29_DIAG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

2.3.5.1 FLTEMU_CONFIG Register (Offset = 0h) [Reset = 00000000h]

FLTEMU_CONFIG is shown in Figure 2-32 and described in Table 2-40.

Return to the Summary Table.

Fault emulation configuration register

Figure 2-32. FLTEMU_CONFIG Register

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

Table 2-40. FLTEMU_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	KEY	W	0h	Write Key In order to write to any bit in this register, 0xA5 must be written to this field at the same time. Otherwise, writes are ignored. The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
7-2	RESERVED	R	0h	Reserved
1	DBL_BIT_INJ_EN	R/W	0h	0' : Fault emulation on data buses (chosen using [FLTEMU_ACCGRPSEL]DATA_GROUP_SEL) happens on the data bit field = [FLTEMU]BITSEL '1' : Fault emulation on data buses (chosen using [FLTEMU_ACCGRPSEL]DATA_GROUP_SEL) happens on the data bit fields [FLTEMU]BITSEL and [FLTEMU]BITSEL + 1 Note: It should be ensured that both data bits where fault injection happens belong to same data checker. For example, if [FLTEMU]BITSEL = 31 and a 32 bit check is being done on the data, then uncorrectable error is not generated as bit31 and bit32 are not checked in the same 32b checker and so a correctable error is generated instead Reset type: SYSRSn
0	ENABLE	R/W	0h	1' : Fault emulation enable '0' : Fault emulation disable. When this bit is '0', the fault injection is disabled. To enable error indication when this bit is '1', SIC_CONFIG.E2E_EN has to be set to '1' Reset type: SYSRSn

2.3.5.2 FLTEMU_ACCGRPSEL Register (Offset = 4h) [Reset = 00000000h]

FLTEMU_ACCGRPSEL is shown in Figure 2-33 and described in Table 2-41.

Return to the Summary Table.

Fault emulation access information group selection register

Figure 2-33. FLTEMU_ACCGRPSEL Register

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

Table 2-41. FLTEMU_ACCGRPSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	DATA_GROUP_SEL	R/W	0h	00 : Reserved 01: Program fetch data group 02: Data Read1 data group 04: Data Read2 data group 08: Data Write data group All other values are reserved Note: To avoid spurious error injections, CTRL_GROUP_SEL field is recommended to be kept zero for Data group fault emulation. Reset type: SYSRSn
7-0	CTRL_GROUP_SEL	R/W	0h	00 : Reserved 01: Program fetch control group 02: Data Read1 control group 04: Data Read2 control group 08: Data Write control group All other values are reserved Note: To avoid spurious error injections, DATA_GROUP_SEL field is recommended to be kept zero for CTRL group fault emulation. Reset type: SYSRSn

2.3.5.3 FLTEMU_BITSEL Register (Offset = 8h) [Reset = 00000000h]

FLTEMU_BITSEL is shown in Figure 2-34 and described in Table 2-42.

Return to the Summary Table.

Fault emulation bitsel

Figure 2-34. FLTEMU_BITSEL Register

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

Table 2-42. FLTEMU_BITSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6-0	BITSEL	R/W	0h	For Control group, this field is interpreted as follows: 0: Fault injection happens on ACK received from chip resource 1 : Fault injection happens on ADDR bit 0 2 : Fault injection happens on ADDR bit 1 .. 32 : Fault injection happens on ADDR bit 31 33 : Fault injection happens on BYTEEN bit 0 34: Fault injection happens on BYTEEN bit 1 .. 40 : Fault injection happens on BYTEEN bit 7 41 : Fault injection happens on BYTEEN bit 8 (only applicable for program fetch access) .. 48 : Fault injection happens on BYTEEN bit 15 (only applicable for program fetch access) 49 : Fault injection happens on SIZE bit 0 50 : Fault injection happens on SIZE bit 1 .. 53 : Fault injection happens on SIZE bit 4 (only applicable for program fetch access) 54 : Reserved 55: Fault injection happens on READY received from chip resource ---------- For data group, this field is interpreted as follows: 0: Fault injection happens on data bit 0 1: Fault injection happens on data bit 1 2: Fault injection happens on data bit 2 .. 63: Fault injection happens on data bit 63 64: Fault injection happens on data bit 64 (applicable only for fetch access) 65: Fault injection happens on data bit 65 (applicable only for fetch access) .. 127: Fault injection happens on data bit 127 (applicable only for fetch access) Reset type: SYSRSn

2.3.5.4 FLTEMU_ADDR Register (Offset = Ch) [Reset = 00000000h]

FLTEMU_ADDR is shown in Figure 2-35 and described in Table 2-43.

Return to the Summary Table.

Fault emulation access address register

Figure 2-35. FLTEMU_ADDR Register

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

Table 2-43. FLTEMU_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R/W	0h	Fault emulation is performed for accesses with this address Reset type: SYSRSn

2.3.5.5 TMU_ROM_PAR_FORCE Register (Offset = 10h) [Reset = 00000000h]

TMU_ROM_PAR_FORCE is shown in Figure 2-36 and described in Table 2-44.

Return to the Summary Table.

Force TMU ROM Parity error

Figure 2-36. TMU_ROM_PAR_FORCE Register

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

Table 2-44. TMU_ROM_PAR_FORCE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	KEY	W	0h	Write Key In order to write to any bit in this register, 0xA5 must be written to this field at the same time. Otherwise, writes are ignored. The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
7-1	RESERVED	R	0h	Reserved
0	TMU_ROM_PAR_FORCE	R/W	0h	1' : TMU ROM Parity force erroe '0' : NA Reset type: SYSRSn

2.3.6 C29_SELFTEST_REGS Registers

Table 2-45 lists the memory-mapped registers for the C29_SELFTEST_REGS registers. All register offset addresses not listed in Table 2-45 should be considered as reserved locations and the register contents should not be modified.

Table 2-45. C29_SELFTEST_REGS Registers

Offset	Acronym	Register Name	Protection
0h	SELFTEST_DIAG_DATA0	Diagnostics data register 0	
4h	SELFTEST_DIAG_DATA1	Diagnostics data register 1	
8h	SELFTEST_DIAG_DATA2	Diagnostics data register 2	
20h	SELFTEST_DIAG_ECC	Diagnostics ECC	
28h	SELFTEST_DIAG_CONTROL	Diagnostic test enable	
2Ch	SELFTEST_DIAG_STATUS	Diagnostic status register	
30h	SELFTEST_DIAG_STATUS_CLR	Diagnostic status clear register	

Complex bit access types are encoded to fit into small table cells. Table 2-46 shows the codes that are used for access types in this section.

Table 2-46. C29_SELFTEST_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

2.3.6.1 SELFTEST_DIAG_DATA0 Register (Offset = 0h) [Reset = 00000000h]

SELFTEST_DIAG_DATA0 is shown in Figure 2-37 and described in Table 2-47.

Return to the Summary Table.

Diagnostics data register 0

Figure 2-37. SELFTEST_DIAG_DATA0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SELFTEST_DIAG_DATA0																															
R/W-0h																															

Table 2-47. SELFTEST_DIAG_DATA0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SELFTEST_DIAG_DATA0	R/W	0h	Self test Diagnostics data 0 This register is used to specify the [31:0] bits of the data to perform Self test ECC checker diagnostics. Reset type: SYSRSn

2.3.6.2 SELFTEST_DIAG_DATA1 Register (Offset = 4h) [Reset = 00000000h]

SELFTEST_DIAG_DATA1 is shown in Figure 2-38 and described in Table 2-48.

Return to the Summary Table.

Diagnostics data register 1

Figure 2-38. SELFTEST_DIAG_DATA1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SELFTEST_DIAG_DATA1																															
R/W-0h																															

Table 2-48. SELFTEST_DIAG_DATA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SELFTEST_DIAG_DATA1	R/W	0h	Self test Diagnostics data 1 This register is used to specify the [63:32] bits of the data to perform Self test ECC checker diagnostics. Reset type: SYSRSn

2.3.6.3 SELFTEST_DIAG_DATA2 Register (Offset = 8h) [Reset = 00000000h]

SELFTEST_DIAG_DATA2 is shown in Figure 2-39 and described in Table 2-49.

Return to the Summary Table.

Diagnostics data register 2

Figure 2-39. SELFTEST_DIAG_DATA2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SELFTEST_DIAG_DATA2																															
R/W-0h																															

Table 2-49. SELFTEST_DIAG_DATA2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SELFTEST_DIAG_DATA2	R/W	0h	Self test Diagnostics data 2 This register is used to specify the [95:64] bits of the data to perform Self test ECC checker diagnostics. Reset type: SYSRSn

2.3.6.4 SELFTEST_DIAG_ECC Register (Offset = 20h) [Reset = 00000000h]

SELFTEST_DIAG_ECC is shown in Figure 2-40 and described in Table 2-50.

Return to the Summary Table.

Diagnostics ECC

Figure 2-40. SELFTEST_DIAG_ECC Register

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

Table 2-50. SELFTEST_DIAG_ECC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	SELFTEST_DIAG_ECC	R/W	0h	Self test Diagnostics ECC This register is used to specify the ECC to perform Self test ECC checker diagnostics Reset type: SYSRSn

2.3.6.5 SELFTEST_DIAG_CONTROL Register (Offset = 28h) [Reset = 60080000h]

SELFTEST_DIAG_CONTROL is shown in Figure 2-41 and described in Table 2-51.

Return to the Summary Table.

Enable diagnostic test

Figure 2-41. SELFTEST_DIAG_CONTROL Register

31	30	29	28	27	26	25	24
DIAG_DATA_WIDTH							
R/W-60h							
23	22	21	20	19	18	17	16
RESERVED				DIAG_ECC_WIDTH			
R-0h				R/W-8h			
15	14	13	12	11	10	9	8
RESERVED				DIAG_CHECKER_SEL			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED	DIAG_SAFETY_SEL	DIAG_ADDITIONAL_PIPELINE_EN	DIAG_MICRO_STEP_MODE	DIAG_TEST_EN			
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h			

Table 2-51. SELFTEST_DIAG_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DIAG_DATA_WIDTH	R/W	60h	ECC data width - Maximum configurable width is 96 (0x60) Reset type: SYSRSn
23-20	RESERVED	R	0h	Reserved
19-16	DIAG_ECC_WIDTH	R/W	8h	ECC bit width - Maximum configurable width is 8 (0x8) Reset type: SYSRSn
15-12	RESERVED	R	0h	Reserved
11-8	DIAG_CHECKER_SEL	R/W	0h	This field is used to select the ECC checker. 0000 : IHW ECC checker 0001 : Interrupt vector ECC checker 0010 : Fetch unit 64 bit ECC checker 0011 : Fetch unit 32 bit ECC checker 0100 : Fetch unit 16 bit ECC checker 0101 : Exe unit DR1 64 bit ECC checker 0110 : Exe unit DR1 32 bit ECC checker 0111 : Exe unit DR1 16 bit ECC checker 1000 : Exe unit DR2 64 bit ECC checker 1001 : Exe unit DR2 32 bit ECC checker 1010 : Exe unit DR2 16 bit ECC checker Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6	DIAG_SAFETY_SEL	R/W	0h	Diagnostic safety selection 0 - ECC 1 - Parity Reset type: SYSRSn

Table 2-51. SELFTEST_DIAG_CONTROL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	DIAG_ADDITIONAL_PIPELINE_EN	R/W	0h	Enable additional pipeline Self test module FSM is looking the error signals after 3 cycles from diag data generation. 1 pipeline implemented inside self test module. Another pipeline should be implemented outside self test module. 3rd pipeline can be implemented outside self test module based on the timing requirements, else user needs to enable this bit to maintain 3 cycles. 1 : Enable additional pipeline 0 : disable additional pipeline Additional pipeline should be enabled for 'IHW ECC checker' and 'Interrupt vector ECC checker' modules. Reset type: SYSRSn
4	DIAG_MICRO_STEP_MODE	R/W	0h	Enable microstep diagnostic mode This field is valid only when DIAG_TEST_EN = 1100 1 : micro step mode 0 : full mode Reset type: SYSRSn
3-0	DIAG_TEST_EN	R/W	0h	Enable self test mechanism 0011 : Enable self test 1100 : Enable self test if i_des_diag_hw_self_test_trig(HW_SELF_TEST_EN) signal is '1' This field will be '0000' once test done. User needs to write into this register again for next test. Reset type: SYSRSn

2.3.6.6 SELFTEST_DIAG_STATUS Register (Offset = 2Ch) [Reset = 00000000h]

SELFTEST_DIAG_STATUS is shown in Figure 2-42 and described in Table 2-52.

Return to the Summary Table.

Diagnostic status

Figure 2-42. SELFTEST_DIAG_STATUS Register

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

Table 2-52. SELFTEST_DIAG_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	DIAG_FAIL_BIT_INDEX	R	0h	This field is used to specify the position of the flipped bit when test failed. For 2 bit flips, this field points the bit position of the first bit. The second bit will be always adjacent to the first bit. Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6-5	DIAG_FAIL_CHECK_TYPE	R	0h	00 : Positive check 01 : Flips one bit 10 : Flips two bit 11 : Reserved Reset type: SYSRSn
4	DIAG_FAIL_UC_ERROR	R	0h	This field is used to specify the diagnostic uncorrectable error when Test failed. Reset type: SYSRSn
3	DIAG_FAIL_C_ERROR	R	0h	This field is used to specify the diagnostic correctable error when Test failed. Reset type: SYSRSn
2	DIAG_TEST_FAIL	R	0h	1 : Test failed (Unexpected error events(C_ERROR/UC_ERROR) occured during self test) 0 : Test passed Reset type: SYSRSn
1	DIAG_TEST_DONE	R	0h	Completed self test. Reset type: SYSRSn
0	DIAG_MICROSTEP_DONE	R	0h	Completed Micro step (16 cycles). HW will clear this bit automatically when the next microstep triggered. Reset type: SYSRSn

2.3.6.7 SELFTEST_DIAG_STATUS_CLR Register (Offset = 30h) [Reset = 00000000h]

SELFTEST_DIAG_STATUS_CLR is shown in Figure 2-43 and described in Table 2-53.

Return to the Summary Table.

Diagnostic status clear

Figure 2-43. SELFTEST_DIAG_STATUS_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					DIAG_TEST_FAIL	DIAG_TEST_DONE	DIAG_MICROSTEP_DONE
R-0h					R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 2-53. SELFTEST_DIAG_STATUS_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	DIAG_TEST_FAIL	R-0/W1S	0h	Clear Test failed status flags 0: Writing a 0 has no effect. 1: Writing a 1 will clear the fields SELFTEST_DIAG_STATUS[DIAG_TEST_FAIL], SELFTEST_DIAG_STATUS[DIAG_FAIL_C_ERROR], SELFTEST_DIAG_STATUS[DIAG_FAIL_UC_ERROR], SELFTEST_DIAG_STATUS[DIAG_FAIL_CHECK_TYPE], SELFTEST_DIAG_STATUS[DIAG_FAIL_BIT_INDEX]. Reset type: SYSRSn
1	DIAG_TEST_DONE	R-0/W1S	0h	Clear selftest done status flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the SELFTEST_DIAG_STATUS[DIAG_TEST_DONE] bit. Reset type: SYSRSn
0	DIAG_MICROSTEP_DONE	R-0/W1S	0h	Clear micro step done status flag. 0: Writing a 0 has no effect. 1: Writing a 1 will clear the SELFTEST_DIAG_STATUS[DIAG_MICROSTEP_DONE] bit. Reset type: SYSRSn

Chapter 3

System Control and Interrupts

This chapter explains system control and interrupts for the C29x cores found on this MCU. The system control module configures and manages the overall operation of the device, and provides information about the device status. The details relating to system control include various sections like reset control, bus architecture, power control, clock control, low-power modes and memory controller subsystem.

3.1 C29x System Control Introduction

On this device, the CPU1 subsystem acts as a controller, and by default (upon reset), the CPU1 subsystem owns all the configuration and control. Through software running on CPU1, peripherals and I/Os can be configured to be accessible by the other CPU subsystems and the chosen configurations can be locked.

The PLL clock configuration is also owned by the CPU1 subsystem by default and HSM subsystem can also access clock configuration if permissions are enabled (by default enabled).

Each CPU can be independently configured to accept interrupts from different peripherals. The interrupt path is divided into three stages – the peripheral (in some cases ESM also), the PIPE, and the CPU. All stages must be configured and enabled for an interrupt to propagate to the CPU.

Each CPU has their own NMI module to handle different exceptions during run time. If the NMI was on CPU1, any NMI exception that is not handled before the NMI Watchdog (ESM NMIWD) timer expiration resets the entire device using XRSn. If the NMI was on any other CPU subsystem, then the specific CPU subsystem is reset (or the entire device is reset depending on the configuration) and all other CPU subsystems can be informed using ESM that the CPU subsystem was reset because of NMIWD timer expiration.

Each CPU subsystem has their own watchdog timer module for software to use. Watchdog timer expiration on CPU1 resets the entire device and Watchdog timer expiration on other CPU resets that specific CPU subsystem alone when configured to generate a reset.

The register space of the device system control module can be found in Section 3.13.

This chapter explains the system control module on all the CPU subsystems.

3.2 System Control Functional Description

The system control module provides the following capabilities:

- Device identification and configuration registers
- Reset control
- Exceptions and Interrupt control (Refer to PIPE Chapter)
- Safety and error handling features of the device
- Power control
- Clock control
- Low Power modes
- Bus Architecture
- Memory Controller Subsystem

3.2.1 Device Identification

Device identification registers provide information on device class, device family, revision, part number, pin count, and device qualification status.

All of the device information is part of the DEV_CFG_REGS space and is accessible only by the software running on the CPU1 subsystem.

The F29x device identification registers are: PARTIDL, PARTIDH, and REVID.

The Unique ID (UID) is available in SOCID information stored at M0 RAM location 0x2000_0980.

3.2.2 Device Configuration Registers

Several registers provide users with configuration information for the C29x subsystems for debug and identification purposes. This information includes the features of the peripherals and how much RAM and Flash memory is available on this part.

These registers are part of DEV_CFG_REGS space and are accessible only by the software running on the CPU1 subsystem.

- MCUCNFx: MCU Configuration registers have been implemented to enable customers to emulate a sub-set device on a super-set device. The following function is provided to configure the device accordingly.
  - `SysCtl_emulateDevice( SysCtl_DevicePartNumber partNumber )`
- CPUID: CPU identification register. This register is available for software to identify on which CPU the register is executing.

3.3 Resets

This section explains the types and effects of the different resets on this device.

3.3.1 Reset Sources

Table 3-1 summarizes the various reset signals and the effect on the device.

Table 3-1. Reset Signals

Reset Source	LPOST	HSM Reset	CPU1 Subsystem Reset	CPU2 Subsystem Reset	CPU3 Subsystem Reset	JTAG / Debug Logic Reset	IOs	XRSn Output
PORESETn_RAW	Yes	Yes	Yes	Yes	Yes	Yes	Hi-Z	Yes
PORESETn	-	Yes	Yes	Yes	Yes	Yes	Hi-Z	Yes
XRSn Pin	-	Yes	Yes	Yes	Yes	-	Hi-Z	-
CPU1.SIMRESET.XRSn	-	Yes	Yes	Yes	Yes	-	Hi-Z	Yes
CPU1.WDRSn	-	Yes	Yes	Yes	Yes	-	Hi-Z	Yes
ESM CPU1.NMIWDRSn(1)	-	Yes	Yes	Yes	Yes	-	Hi-Z	Yes
CPU1.SYSRSn (Debugger Reset)	-	-	Yes	Yes	Yes	-	Hi-Z	-
CPU2.WDRSn	-	-	-	Yes	-	-	-	-
ESM CPU2.NMIWDRSn(1)	-	Yes	Yes	Yes	Yes	-	Hi-Z	Yes
CPU2.SYSRSn (Debugger Reset)	-	-	-	Yes	-	-	-	-
CPU3.WDRSn	-	-	-	-	Yes	-	-	-
ESM CPU3.NMIWDRSn(1)	-	Yes	Yes	Yes	Yes	-	Hi-Z	Yes
CPU3.SYSRSn (Debugger Reset)	-	-	-	-	Yes	-	-	-
ECAT_RESET_OUT	-	Yes	Yes	Yes	Yes	-	Hi-Z	Yes

(1) Applicable only if ESM CPU instances are programmed in ESMXRSNCTL register to trigger XRSn. Refer to Chapter 7 for more details.

Note

After a reset, the reset cause register (RESC) is updated with the reset cause. The bits in this register maintain the state across multiple resets. These can only be cleared by a power-on reset (POR) or by writing 1 to the corresponding bit in RESCCLR register (status can be cleared by also writing a 1 to the RESC register bits). Each CPU has an RESC register, referred to as CPU1.RESC, CPU2.RESC and CPU3.RESC.

The resets can be divided into a few groups:

- Chip-level resets (XRSn, PORESETn, CPU1.WDRSn, ESM CPU1.NMIWDRSn (if enabled to trigger XRSn), ESM CPU2.NMIWDRSn (if enabled to trigger XRSn), ESM CPU3.NMIWDRSn (if enabled to trigger XRSn), SIMRESET.XRSn and ECAT_RESET_OUT (if enabled)), which reset all or almost all of the device.
- System resets (CPU1.$\overline{\text{SYSRS}}$), which reset a large subset of the device but maintain some system-level configuration.
- CPU2 subsystem resets (CPU2. SYSRSn, CPU2. WDRSn, ESM CPU2. NMIWDRSn (if configured to trigger CPU3.RSn)), which reset only CPU2 and the peripherals.
- CPU3 subsystem resets (CPU3. SYSRSn, CPU3. WDRSn, ESM CPU3. NMIWDRSn (if configured to trigger CPU3.RSn)), which reset only CPU3 and the peripherals.

Whenever the CPU1 subsystem is reset, CPU2 and CPU3 also gets reset if configured in SSU CPU2 and CPU3 Reset Control (CPU_RST_CTRL) registers and held in reset until CPU1 brings CPU2 and CPU3 out of reset by writing to the CPU_RST_CTRL register or as determined by HSM input. This is done by user application code on CPU1.

Many peripheral modules have individual resets accessible through the system control registers. For information about a module reset state, refer to the appropriate chapter for that module.

Note

After a POR, the boot ROMs clear all of the system and message RAMs on all CPUs.

Figure 3-1 captures the overall device level reset connectivity and propagation.

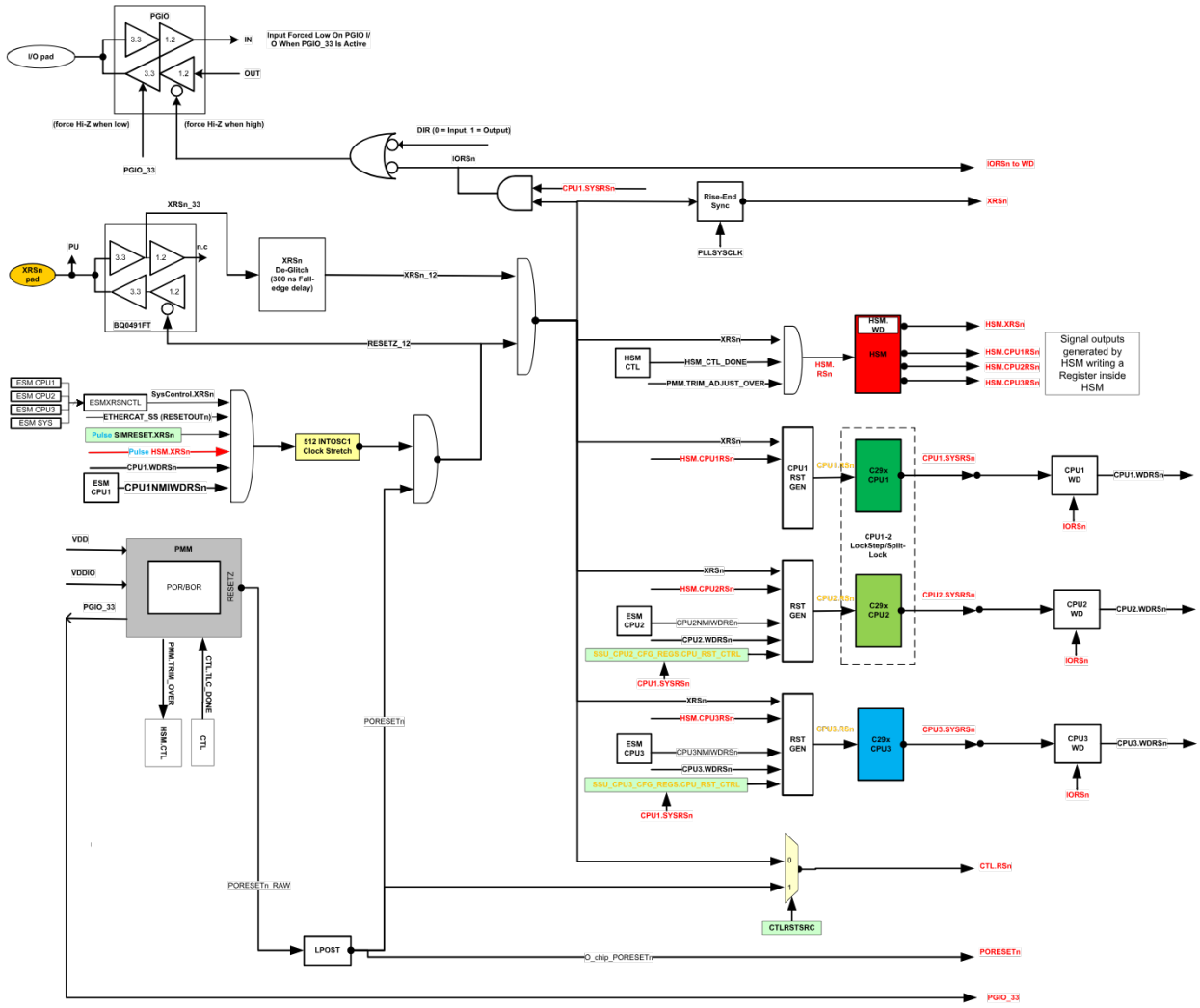


Figure 3-1. Device Reset Diagram

3.3.2 External Reset ($\overline{XRS}$)

The external reset ($\overline{XRS}$) is the main chip-level reset for the device. $\overline{XRS}$ resets all C29x CPUs, HSM , all peripherals and I/O pin configurations, and most of the system control registers. $\overline{XRS}$ also holds CPU2, CPU3 and HSM in reset. There is a dedicated open-drain pin for $\overline{XRS}$. This pin can be used to drive reset pins for other ICs in the application, and can be driven by an external source. The $\overline{XRS}$ is driven internally during watchdog, NMI, and power-on resets.

The XRSn bit in the RESC register is set whenever $\overline{XRS}$ is driven low for any reason. This bit is then cleared by the boot ROM.

3.3.3 Simulate External Reset

The user can simulate an external reset ($\overline{XRS}$) by setting the XRSn bit to 1 in the SIMRESET register using CPU1 software. This toggles the $\overline{XRS}$ pin; hence, resetting the full device (just like an external reset).

After this reset, the SIMRESET_XRSn bit in the RESC register is set. Software can read this bit to know the cause of the reset and clear the status by writing a 1 into the corresponding bit in the RESCCLR register.

3.3.4 Power-On Reset (POR)

The power-on reset (POR) circuit creates a clean reset throughout the device during power-up, suppressing glitches on the GPIOs. The $\overline{\text{XRS}}$ pin is held low for the duration of the POR. In most applications, $\overline{\text{XRS}}$ is held low long enough to reset other system ICs, but some applications can require a longer pulse. In these cases, $\overline{\text{XRS}}$ can be driven low externally to provide the correct reset duration. A POR resets everything that $\overline{\text{XRS}}$ does, along with the register – the reset cause register (RESC).

After a POR, the POR and XRSn bits in RESC are set. These bits are then cleared by the boot ROM.

3.3.5 Debugger Reset ($\overline{\text{SYSRS}}$)

During development, it is sometimes necessary to reset the CPU and the peripherals without disconnecting the debugger or disrupting the system-level configuration. To facilitate this, each CPU has a subsystem reset, which can be triggered by a debugger using Code Composer Studio IDE. CPU2 subsystem reset (CPU2.$\overline{\text{SYSRS}}$) resets only CPU2, the peripherals, and the clock gating and LPM configuration. It does not hold CPU2 in reset. CPU3 subsystem reset (CPU3.$\overline{\text{SYSRS}}$) resets only CPU3, the peripherals, and the clock gating and LPM configuration. CPU1 subsystem reset (CPU1.$\overline{\text{SYSRS}}$) resets CPU1, the peripherals, many system control registers (including the clock gating and LPM configuration and the peripheral CPU ownership), and all I/O pin configurations. CPU1.$\overline{\text{SYSRS}}$ also produces a CPU2.$\overline{\text{SYSRS}}$ and CPU3.$\overline{\text{SYSRS}}$ (CCS Gel file may have code to release CPU2 and CPU3 out of reset on CPU1 debug reset).

Figure 3-2 explains how the all CPU $\overline{\text{SYSRS}}$ and Software Reset signals are connected to cause particular peripheral to reset in the device.

Neither $\overline{\text{SYSRS}}$ resets the debug module, the device capability registers, the clock source and PLL configurations, the missing clock detection state, the PIPE vector fetch error handler address, the NMI flags, the analog trims, or anything reset only by a POR (see Section 3.3.4).

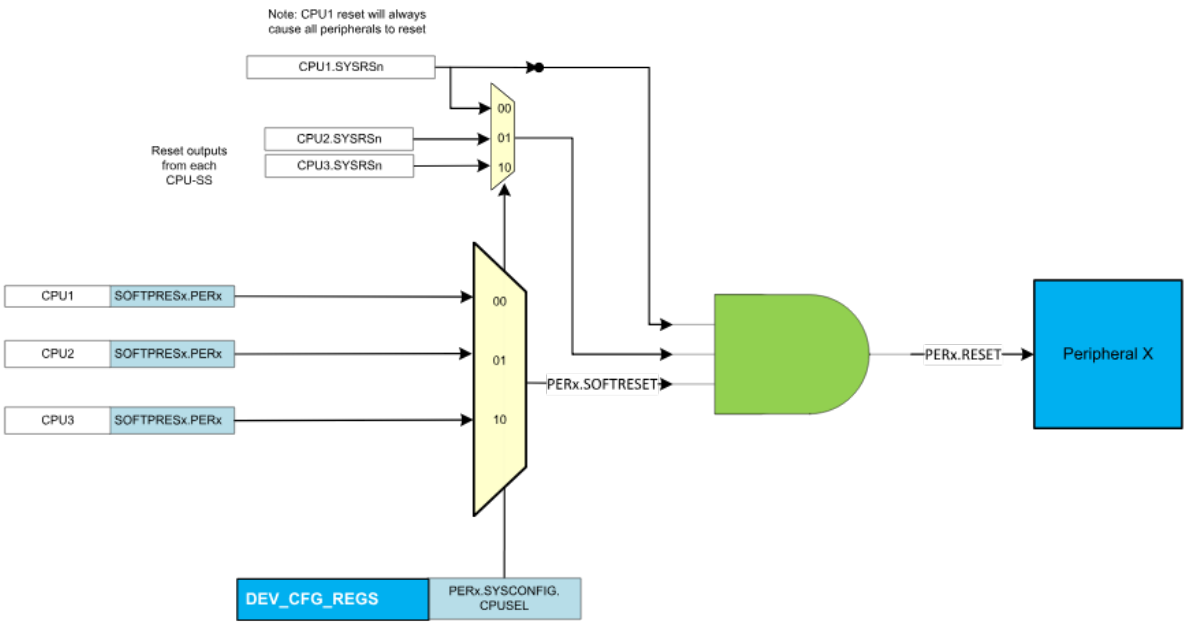


Figure 3-2. Software and SYSRSn Connectivity Diagram for Peripheral Reset

Note

Since the reset signals are active low any input to the AND gate when active (low) outputs the peripheral reset signal to be active (low).

3.3.6 Watchdog Reset ($\overline{\text{WDRS}}$)

Each CPU has a watchdog timer that can optionally trigger a reset that lasts for 512 INTOSC1 cycles. CPU1 watchdog reset (CPU1.$\overline{\text{WDRS}}$) produces an $\overline{\text{XRS}}$. CPU2 watchdog reset (CPU2.$\overline{\text{WDRS}}$) produces a CPU2.$\overline{\text{SYSRS}}$ and similarly CPU3 watchdog reset (CPU3.$\overline{\text{WDRS}}$) produces a CPU3.$\overline{\text{SYSRS}}$.

After a watchdog reset, the WDRSn bit in the RESC register is set. Software can read this bit to know the cause of reset and clear the status by writing a 1 into the corresponding bit in the RESCCLR register.

3.3.7 ESM NMI Watchdog Reset ($\overline{\text{NMIWDRS}}$)

Each ESM CPU module in ESM Subsystem (ESM-SS) has a watchdog timer that triggers a reset if the CPU does not respond to an error within a user-specified amount of time. ESM CPU1 NMI watchdog reset (ESM CPU1.$\overline{\text{NMIWDRS}}$) produces an $\overline{\text{XRS}}$. ESM CPU2 NMI watchdog reset (CPU2.$\overline{\text{NMIWDRS}}$) produces a CPU2.$\overline{\text{SYSRS}}$ and ESM CPU3 NMI watchdog reset (CPU3.$\overline{\text{NMIWDRS}}$) produces a CPU3.$\overline{\text{SYSRS}}$.

After an NMI watchdog reset, the NMIWDRSn bit in the RESC register is set.

3.3.8 EtherCAT Slave Controller (ESC) Module Reset Output

You can configure the EtherCAT Slave Controller (ESC) module to drive the $\overline{\text{XRS}}$ pin low whenever the ESC module receives a reset. This is done by setting the DEVICE_RESET_EN bit to 1 in the ESCSS_RESET_DEST_CONFIG register of the ESC module (EtherCAT subsystem). By default, this is not enabled. Since this toggles the $\overline{\text{XRS}}$ pin, all effects of an external $\overline{\text{XRS}}$ reset take effect.

After an ECAT_RESET_OUT reset, the ECAT_RESET_OUT bit in the RESC register is set. Software can read this bit to know the cause of reset and clear the status by writing a 1 into the corresponding bit in the RESCCLR register.

3.4 Safety Features

This section gives details on features that monitor device operation during run-time to detect any error in operation.

3.4.1 Write Protection on Registers

3.4.1.1 LOCK Protection on System Configuration Registers

Several system configuration registers are protected from spurious CPU writes by “LOCK” registers. Once these associated LOCK register bits are set, the respective locked registers can no longer be modified by software. See the register descriptions for details.

3.4.1.2 EALLOW Protection

EALLOW protection mechanism is not applicable for this device.

3.4.2 PIPE Vector Address Validity Check

PIPE Vector tables are protected by ECC safety mechanism.

3.4.3 NMIWDs

Each ESM CPU instance has user-programmable NMIWD period registers, in which users can set a limit on how much time to allocate for the device to acknowledge the NMI. If the NMI is not acknowledged, the NMI causes a device reset.

3.4.4 System Control Registers Parity Protection

Critical System Control Register are parity protected. The registers that are covered by parity protection have "PARITY" mentioned in the respective "Protection" column in register descriptions table. There is also a provision to test the parity protection feature by enabling parity test using the DEV_CFG_REGS.PARITY_TEST, CPU_PER_CFG_REGS.PARITY_TEST_ALT1 and CPU_SYS_REGS.PARITY_TEST_ALT2 register.

If parity error is detected, ESM is notified and appropriate action is taken based on user configurations. Refer to ESM Chapter for more details.

3.4.5 ECC Enabled RAMs, Shared RAMs Protection

Each CPU subsystem has different RAM blocks. All the RAM blocks are ECC-enabled. All errors go to ESM and all single-bit ECC RAM errors are auto-corrected by ECC logic inside CPU. Also, an interrupt is generated to the corresponding CPU using ErrorAggregator and ESM. Single-bit errors are not corrected inside the RAM blocks; hence, RAM blocks still have the incorrect data (single-bit error). User software needs to write back the correct data.

All uncorrectable double-bit errors end up triggering an NMI to the corresponding CPUs using ErrorAggregator and ESM.

3.4.6 ECC Enabled Flash Memory

Flash single-bit errors are corrected automatically by ECC logic inside CPU, but the Flash single-bit errors are not corrected in Flash memory. Flash memory still contains wrong data until another erase/program operation happens to correct the Flash contents. Irrespective of whether the error interrupt is enabled or disabled, single-bit errors are always corrected. When the interrupt is disabled, users can check the single-bit error flag for any single-bit error occurrences.

Flash uncorrectable errors end up triggering an NMI to the respective CPU using ErrorAggregator and ESM.

3.4.7 ERRORSTS Pin

ERRORSTS signal is driven from Error Signaling Module (ESM). The ERRORSTS signal is ‘always output’ on a selected GPIO pin and remains high until an error is detected inside the chip. On an error, the ERRORSTS pin goes low (default polarity) until the corresponding internal error status flag for that error source is cleared.

The ERRORSTS pin is tri-stated until the chip power rails ramp up to the lower operational limit. As the ERRORSTS pin is an active-low pin (default polarity), users who care about the state of this pin during power-up can connect an external pull-down on this pin. Polarity of ERRORSTS pin is configurable (default setting is active-low polarity).

3.5 Clocking

This section explains the clock sources and clock domains on this device, and how to configure them for application use. Figure 3-3 provides an overview of the device clocking system.

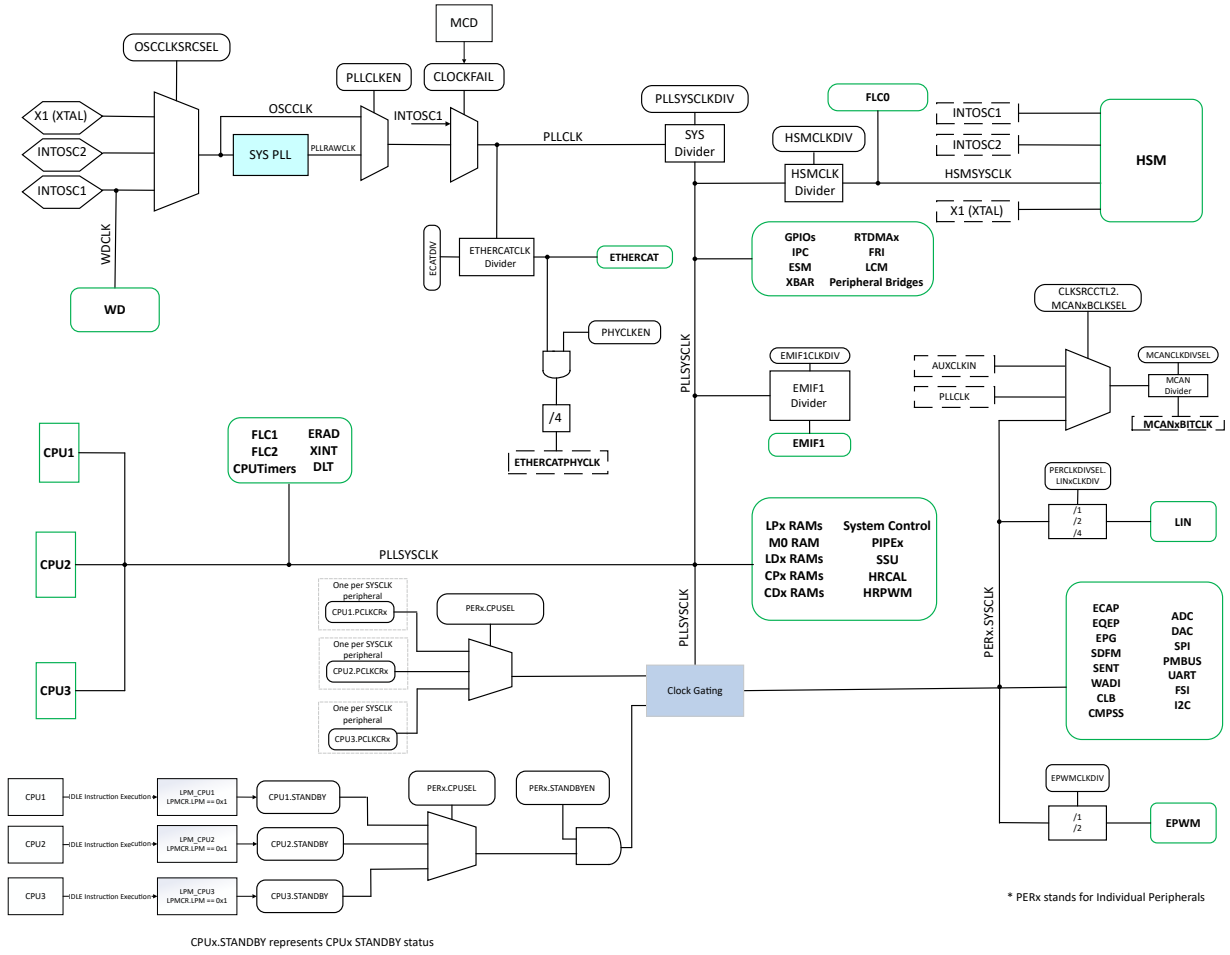


Figure 3-3. Clocking System

3.5.1 Clock Sources

The needs of the application are what ultimately determine the clock configuration. Specific concerns such as application performance, power consumption, total system cost, and EMC are beyond the scope of this document, but consider the following questions:

1. What is the desired CPU frequency?
2. Are there any additional communication protocols or peripherals required ?
3. What types of external oscillators or clock sources are available?

If required an external clock source with a precise frequency must be used as a reference clock. Otherwise, it may be possible to use only INTOSC2 and avoid the need for more external components.

All of the clocks in the device are derived from one of four clock sources.

3.5.1.1 Primary Internal Oscillator (INTOSC2)

At power-up, the device is clocked from an on-chip 10MHz oscillator (INTOSC2). INTOSC2 is the primary internal clock source and is the default system clock at reset. INTOSC2 is used to run the boot ROM and can be used as the system clock source for the application.

Note

INTOSC2 frequency tolerance is too loose to meet the timing requirements for some communication peripherals, so an external clock must be used to support those features.

3.5.1.2 Backup Internal Oscillator (INTOSC1)

The device also includes a redundant on-chip 10 MHz oscillator (INTOSC1). INTOSC1 is a backup clock source that normally only clocks the watchdog timers and missing clock detection circuit (MCD). If MCD is enabled and a missing system clock is detected, the system PLL is bypassed and all system clocks are connected to INTOSC1 automatically. INTOSC1 may also be manually selected as the system and auxiliary clock source for debug purposes.

3.5.1.3 External Oscillator (XTAL)

The dedicated X1 and X2 pins support an external clock source (XTAL), which can be used as the main system. Frequency limits and timing requirements can be found in the device data sheet. Three types of external clock sources are supported:

- A single-ended 3.3V external clock. The clock signal must be connected to X1 while X2 is left unconnected, as shown in Figure 3-4.

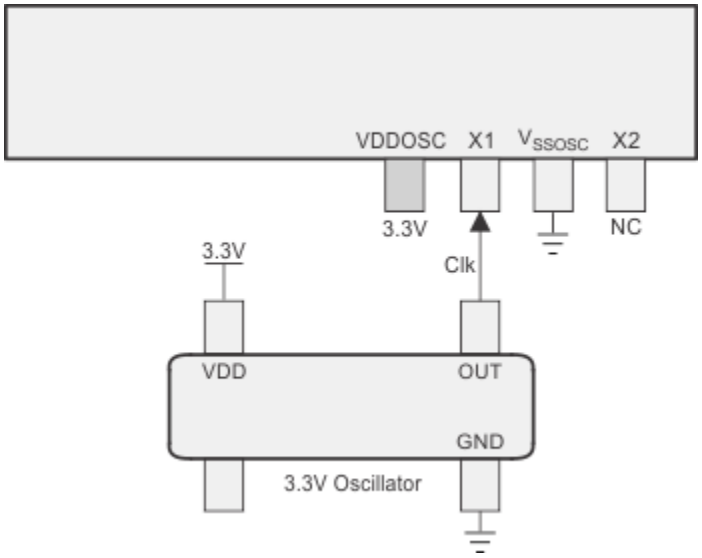


Figure 3-4. Single-ended 3.3V External Clock

- An external crystal. The crystal must be connected across X1 and X2 with the load capacitors connected to VSSOSC as shown in Figure 3-5.

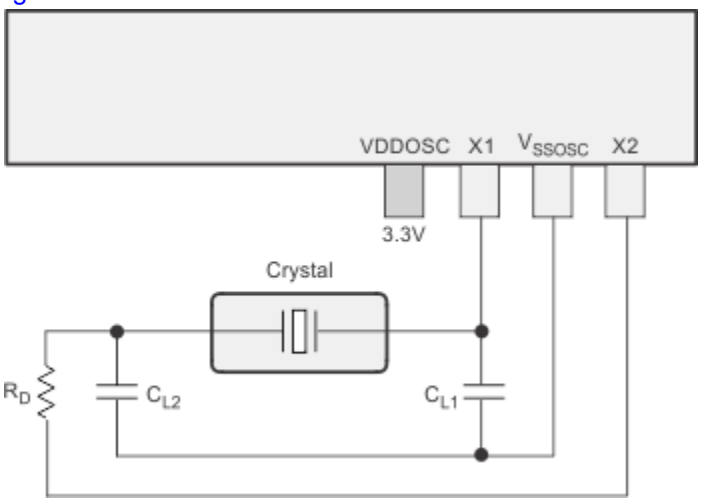


Figure 3-5. External Crystal

- An external resonator. The resonator must be connected across X1 and X2 with the ground connected to VSSOSC as shown in Figure 3-6.

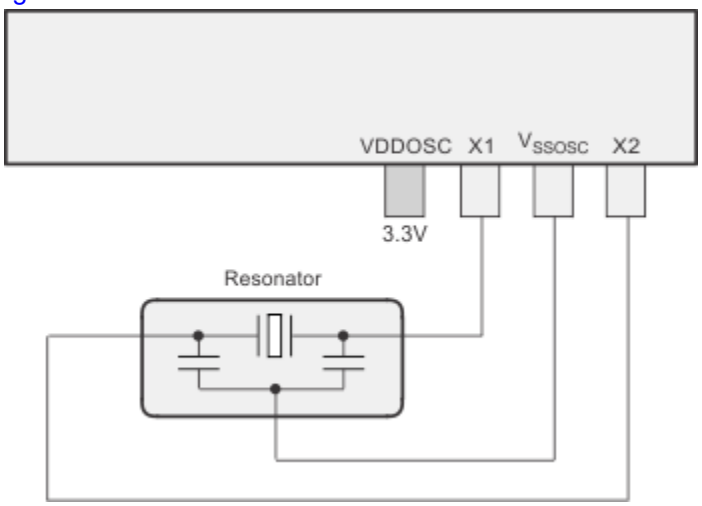


Figure 3-6. External Resonator

3.5.1.4 Auxiliary Clock Input (AUXCLKIN)

An additional external clock source is supported on GPIO133 (AUXCLKIN). This must be a single-ended 3.3V external clock that can be used as the clock source for MCAN bit clock. Frequency limits and timing requirements can be found in the device data sheet. The external clock must be connected directly to the GPIO133 pin, as shown in Figure 3-7.

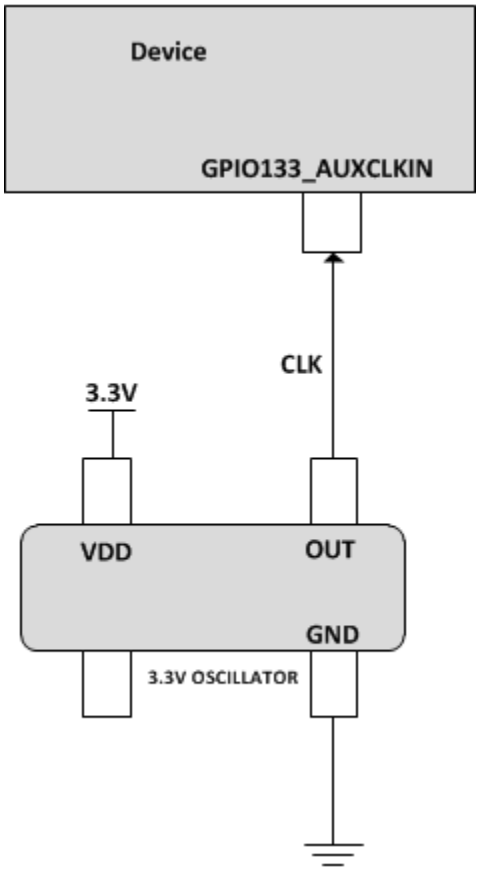


Figure 3-7. Auxiliary Clock Input (AUXCLKIN)

3.5.2 Derived Clocks

The clock sources discussed in the previous section can be multiplied (using PLL) and divided down to produce the desired clock frequencies for the application. This process produces a set of derived clocks, which are described in this section.

3.5.2.1 Oscillator Clock (OSCCLK)

One of INTOSC2, XTAL, or INTOSC1 must be chosen to be the controller reference clock (OSCCLK) for the CPU and most of the peripherals. OSCCLK can be used directly or applied through the system PLL to reach a higher frequency. At reset, OSCCLK is the default system clock and is connected to INTOSC2.

3.5.2.2 System PLL Output Clock (PLLRAWCLK)

The system PLL allows the device to run at the maximum rated operating frequency, and in most applications generates the main system clock. This PLL uses OSCCLK as a reference, and features a fractional multiplier and slip detection. For configuration instructions, see Section 3.5.7.

3.5.3 Device Clock Domains

The device clock domains feed the clock inputs of the various modules in the device. They are connected to the derived clocks, either directly or through an additional divider.

3.5.3.1 System Clock (PLLSYSCLK)

The system control registers, RAMs, IPC module, GPIO qualification, SSU, LCM, Flash read interfaces, FLC1/2 , all SRAMs, XBAR, ESM, Peripheral Bridges, RTDMA1/2 and PIPE modules have a clock domain (PLLSYSCLK). Despite the name, PLLSYSCLK can be connected to the system PLL (PLLRAWCLK) or to OSCCLK. The chosen clock source is run through a frequency divider, which is configured using the SYSCLKDIVSEL register.

3.5.3.2 CPU Clock (CPUCLK)

Each CPU has its own clock (CPU1.CPUCLK,CPU2.CPUCLK and CPU3.CPUCLK) which is used to clock the CPU, and its coprocessor's. This clock is identical to PLLSYSCLK, but is gated when the CPU enters STANDBY mode. CPUCLK provides clock to individual CPU timers (CPUTIMERx) and DLT.

3.5.3.3 Peripheral Clock (PERx.SYSCLK)

This clock is identical to PLLSYSCLK, but is gated when the CPU selected by CPUSEL enters STANDBY mode based on individual STANDBYEN bit in respective peripheral system configuration register (PERxSYSCONFIG).

Each peripheral has a CPU selection logic to select either CPU1, CPU2, or CPU3 which is done by using the CPUSEL register. The CPU selection logic is only for clock gating purposes since there is no CPU to peripheral allocation and any CPU can access and configure any peripheral provided the respective CPU has access to it. Each peripheral clock also has an independent clock gating that is controlled by the CPU PCLKCRx and STANDBYEN registers.

By default, the ePWM, EMIF, and LIN clocks each have an additional /2 divider, which is required to support CPU frequencies over 100MHz. At slower CPU frequencies, these dividers can be disabled using the PERCLKDIVSEL register.

Note

The application needs to wait for 5 SYSCLK cycles after enabling the clock to the peripherals, when using PCLKCRx.

3.5.3.4 MCAN Bit Clock

The required frequency tolerance for the MCAN bit clock depends on the bit timing setup and network configuration, and can be as tight as 0.1%. Since the main system clock (in the form of PERx.SYSCLK) can not be precise enough, the bit clock can also be connected to PLLCLK or AUXCLKIN using the CLKSRCCTL2 register. There is an independent selection for each MCAN bit clock source.

3.5.3.5 CPU Timer2 Clock (TIMER2CLK)

CPU timers 0 and 1 are connected to PLLSYSCLK. Timer 2 is connected to PLLSYSCLK by default, but can also be connected to INTOSC1, INTOSC2, XTAL, FLC1/2 PUMPOSC or CRUDEOSC using the TMR2CLKCTL register. This register also provides a separate prescale divider for timer 2. If a source other than PLLSYSCLK is used, the frequency must be at least twice the source frequency to make sure of correct sampling. Each CPU has independent CPU timers and TMR2CLKCTL register.

The main reason to use a non-SYSCLK source is for internal frequency measurement. In most applications, timer 2 runs off of the SYSCLK.

3.5.4 External Clock Output (XCLKOUT)

To observe a clock directly for debug and testing purposes, the external clock output (XCLKOUT) supports this by connecting a clock to external pins, GPIO73 and GPIO224. The available clock sources are PLLSYSCLK, PLLRAWCLK, PLLCLK, CPUx.CLOCK, XTAL, INTOSC1, and INTOSC2.

To use XCLKOUT, first select the clock source using the CLKSRCCTL3 register. Next, select the desired output divider using the XCLKOUTDIVSEL register. Finally, connect either GPIO73 or GPIO219 to mux channel 3 using the GPIO configuration registers.

3.5.5 Clock Connectivity

Table 3-2 provides details on the clock connections of every module present in the device.

Table 3-2. Clock Connections Sorted by Clock Domain

Clock Domain	CPU1 Subsystem	CPU2 Subsystem	CPU3 Subsystem	Shared Modules
CPUx.CPUCLK	CPU1	CPU2	CPU3	
PLLSYSCLK				PIPEx GPIO Input Sync and Qual IPC XBARs EMIF1 AnalogSubsys SSU System Control Registers HRCAL RTDMAx ESM Peripheral Bridges LCM FRIx, FLC1/2 All SRAMs ERADx XINT HRPWM DLT CPUTimers
PERx.SYSCLK				ADC CMPSS DAC ePWM eCAP eQEP I2C SDFM FSI PMBUS CLB SPI LIN UART SENT WADI EPG

Table 3-2. Clock Connections Sorted by Clock Domain (continued)

Clock Domain	CPU1 Subsystem	CPU2 Subsystem	CPU3 Subsystem	Shared Modules
MCAN Bit Clock				MCAN (CAN-FD)
WDCLK (INTOSC1)	CPU1.Watchdog	CPU2.Watchdog	CPU3.Watchdog	

3.5.6 Using an External Crystal or Resonator

The X1 and X2 pins double as GPIO221 and GPIO220. At power-up, these pins are in GPIO mode and the on-chip crystal oscillator is powered off. The following procedure can be used to switch the pins to X1 and X2 mode and enable the oscillator:

1. Clear the XTALCR.OSCOFF bit.
2. Wait for the crystal to power up. The typical wait time is 1ms, but this depends on the crystal that is being used.
3. Clear the X1 counter by writing a 1 to X1CNT.CLR and keep clearing until the X1 counter value in the X1CNT register is no longer saturated 2047 (0x7FF).
4. Wait for the X1 counter value in the X1CNT register to reach 2047 (0x7FF).
5. Repeat steps 3-4 three additional times.
6. Select XTAL as the OSCCLK source by writing a 1 to CLKSRCCTL1.OSCCLKSRCSEL.
7. Check the MCLKSTS bit in the MCDCR register. If the bit is set, the oscillator has not finished powering up, and more time is required:
   a. Clear the missing clock status by writing a 1 to MCDCR.MCLKCLR.
   b. Repeat steps 2-7. Do not reset the device. Doing so powers down the oscillator, which requires the procedure to be restarted from step 1.
   c. If the oscillator has not finished powering up in 10 milliseconds, there is a real clock failure.
8. If MCDCR.MCLKSTS is clear, the oscillator startup is a success. The system clock is now derived from XTAL.

3.5.6.1 X1/X2 Precondition Circuit

The GPIO220/221 alternate functionality on X1/X2 can be used to speed up the start-up time of the crystal by as much as 30% if needed. This functionality is achieved by preconditioning the load capacitors CL1 and CL2 to a known state before the XTAL is turned on.

The steps below outline the procedure to precondition X1/X2 before turning on the XTAL:

1. DevCfgRegs.XTALCR2.bit.XIF = 1; // Precondition X1 to High
2. DevCfgRegs.XTALCR2.bit.XOF = 1; // Precondition X2 to High
3. DevCfgRegs.XTALCR2.bit.FEN = 1; // Enable X1/X2 Precondition
4. DEVICE_DELAY_US(1);
5. DevCfgRegs.XTALCR.bit.OSCOFF = 0; // Removes Precondition and Turns on the XTAL
6. DevCfgRegs.XTALCR2.bit.FEN = 0; // Disables X1/X2 Precondition

3.5.7 PLL

The PLL is responsible for synthesizing an output frequency from the input clock (from the oscillator). Figure 3-8 shows a simple block diagram of the PLL. The PLL divides the reference input for a lower frequency input into the PLL by (REFDIV + 1). Then multiplies this internal frequency by IMULT to get the VCO output clock. The PLL output is divided by (ODIV + 1) to generate PLLRAWCLK that is further divided by SYSCLKDIVSEL.PLLSYSCLKDIV to generate PLLSYSCLK.

There is PLL also called as SYSPLL and the equations shown in Figure 3-8 can be used to configure the respective PLL.

- IMULT is the integer value of the multiplier
- REFDIV is the reference divider for the OSCCLK
- ODIV is the output divider of the PLLRAWCLK
- PLLSYSCLKDIV is the system clock divider

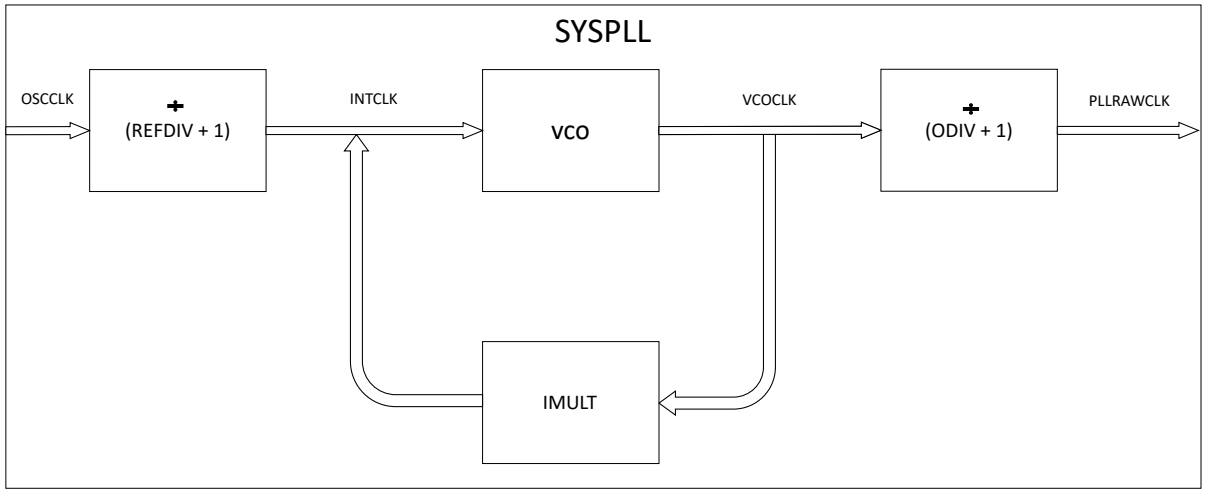
For the permissible values of the multipliers and dividers, see the documentation for their respective registers.

Many combinations of multiplier and divider can produce the same output frequency. However, the product of the reference clock frequency and the multiplier (known as the VCO frequency) must be in the range specified in the data sheet.

Note

The system clock frequency (PLLSYSCLK) cannot exceed the limit specified in the data sheet. This limit does not allow for oscillator tolerance.

The clock source and PLL configuration can only be done by CPU1 and can be read by other CPUs.



$$f_{PLLRAWCLK} = \frac{f\,OSCCLK}{(REFDIV+1)} \times \frac{IMULT}{(ODIV+1)}$$

Figure 3-8. PLL

3.5.7.1 System Clock Setup

Once the application requirements are understood, a specific clock configuration can be determined. The default configuration is for INTOSC2 to be used as the system clock (PLLSYSCLK) with a divider of 1. The following procedure can be used to set up the desired application configuration:

Refer to your device SysCtl_setClock() function inside C2000Ware installation for an example.

Recommended sequence to set up the system PLL:

1. Bypass the PLL by clearing SYSPLLCTL1[PLLCLKEN] and wait for at least 120 CPU clock cycles by adding 120 NOP instructions.
2. Power down the PLL by writing to SYSPLLCTL1.PLLEN = 0 and wait for at least 60 CPU clock cycles by adding 60 NOP instructions.
3. Select the reference clock source (OSCCLK) by writing to CLKSRCCTL1.OSCCLKSRCSEL and wait for at least 300 CPU clock cycles by adding 300 NOP instructions.
4. Set the system clock divider to /1 to make sure of the fastest PLL configuration by clearing SYSCLKDIVSEL[PLLSYSCLKDIV].
5. Set the IMULT, REFDIV, and ODIV simultaneously by writing 32-bit value in SYSPLLMULT at once. This automatically enables the PLL. Be sure the settings for multiplier and dividers do not violate the frequency specifications as defined in the data sheet.
6. Wait for PLL to lock by polling for lock status bit to go high, that is, SYSPLLSTS.LOCKS = 1
7. Configure DCC with reference clock as OSCCLK and clock under measurement as PLLRAWCLK, and verify the frequency of the PLL. If the frequency is out of range, do not enable PLLRAWCLK as SYSCLK, stop here and troubleshoot. Refer to Dual Clock Comparator (DCC) chapter for more information on the configuration and usage.
8. If the PLLRAWCLK is within the valid range, then set the system clock divider one setting higher than the final desired value. For example DEV_CFG_REGS.SYSCLKDIVSEL.bit.PLLSYSCLKDIV = divsel + 1. This limits the current increase when switching to the PLL.
9. Switch to the PLL as the system clock by setting SYSPLLCTL1[PLLCLKEN] and wait for 200 PLLSYSCLK cycles for current to stabilize by adding 200 NOP instructions.
10. Change the system clock divider (PLLSYSCLKDIV) to the appropriate value.

Note

1. SYSPLL must be bypassed and powered down manually before changing the OSCCLK source.
2. At least 120 CPU clock cycles delay is needed after bypassing PLL, that is, SYSPLLCTL1.PLLCLKEN=0.
3. At least 60 CPU clock cycles delay is needed after PLL is powered down, that is, SYSPLLCTL1.PLLEN=0.
4. At least 300 CPU clock cycles delay is needed after OSSCLK source is changed.
5. PLL SLIP bit is not supported. DCC can be used to check the validity of the PLL clock. This feature is included as part of SysCtl_setClock() function inside C2000Ware.

3.5.7.2 SYS PLL Bypass

If the application requires the PLL clock to be bypassed from the system, then the application needs to configure SYSPLLCTL1.PLLCLKEN = 0. It takes up to 120 CPU clock cycles before the bypass is effective. In the meantime, if PLLSYSCLKDIV is reduced to a lower value (for example, from /2 to /1 or /4 to /2), the device can be clocked above the maximum rated frequency and can lead to unpredictable device behavior. Hence, a delay of 120 CPU clock cycles is required after bypassing the PLL from the enable state, that is, going from PLLCLKEN = 1 to PLLCLKEN = 0.

3.5.8 Clock (OSCCLK) Failure Detection

To achieve safety diagnostic, Missing Clock Detection (MCD) can be used. Table 3-3 lists the details.

Table 3-3. Clock Source (OSCCLK) Failure Detection

Clock Failure Detection Circuitry	Clocks Detected	Time for Detection (in Cycles)	Limitations
Missing Clock Detection (MCD)	INTOSC2, XTAL/X1	8192 INTOSC1 cycles	Cannot detect INTOSC1 clock failure.
Dual Clock Comparator (DCC)	INTOSC1, INTOSC2, XTAL/X1	Depends on the frequency of reference and monitored clocks	Software configuration is required for clock monitoring

3.5.8.1 Missing Clock Detection Logic

The missing clock detect (MCD) logic detects OSCCLK failure, using INTOSC1 as the reference clock source. This circuit only detects complete loss of OSCCLK and does not do any detection of frequency drift on the OSCCLK.

This circuit monitors the OSCLK (primary clock) using the 10MHz clock provided by the INTOSC1 (secondary clock) as a backup clock. This circuit functions as:

1. The primary clock (OSCCLK) clock keeps ticking a 7-bit counter (named as MCDPCNT). This counter is asynchronously reset with XRSn.
2. The secondary clock (INTOSC1) clock keeps ticking a 13-bit counter (named as MCDSCNT). This counter is asynchronously reset with XRSn.
3. Each time MCDPCNT overflows, the MCDSCNT counter is reset. Thus, if OSCCLK is present or not slower than INTOSC1 by a factor of 64, MCDSCNT never overflows.
4. If OSCCLK stops for some reason or is slower than INTOSC1 by at least a factor of 64, the MCDSCNT overflows and a missing clock condition is detected on OSCCLK.
5. The above check is continuously active, unless the MCD is disabled using MCDCR register (by making the MCLKOFF bit 1).
6. If the circuit ever detects a missing OSCCLK, the following occurs:
   - The MCDSTS flag is set.
   - The MCDSCNT counter is frozen to prevent further missing clock detection.
   - The CLOCKFAIL signal goes high, which generates TRIP events to PWM modules and can be configured to trigger NMI in Error Signaling Module (ESM)
   - PLL is forcefully bypassed and OSCCLK source is switched to INTOSC1 (System Clock Frequency = INTOSC1 Frequency (10MHz)/SYSDIV). PLLMULT is zeroed out automatically in this case.
   - While the MCDSTS bit is set, the OSCCLKSRCSEL bits have no effect and OSCCLK is forcefully connected to INTOSC1.
   - PLLRAWCLK going to the system is switched to INTOSC1 automatically.
7. If the MCLKCLR bit is written (this is a W = 1 bit), MCDSTS bit is cleared and OSCCLK source is decided by the OSCCLKSRCSEL bits. Writing to MCLKCLR also clears the MCDPCNT and MCDSCNT counters to allow the circuit re-evaluate missing clock detection. If the user wants to lock the PLL after missing clock detection, switch the clock source to INTOSC1 (using OSCCLKSRCSEL register), do a MCLKCLR, and relock the PLL.
8. The MCD is enabled at power up.

Figure 3-9 shows the missing clock logic functional flow.

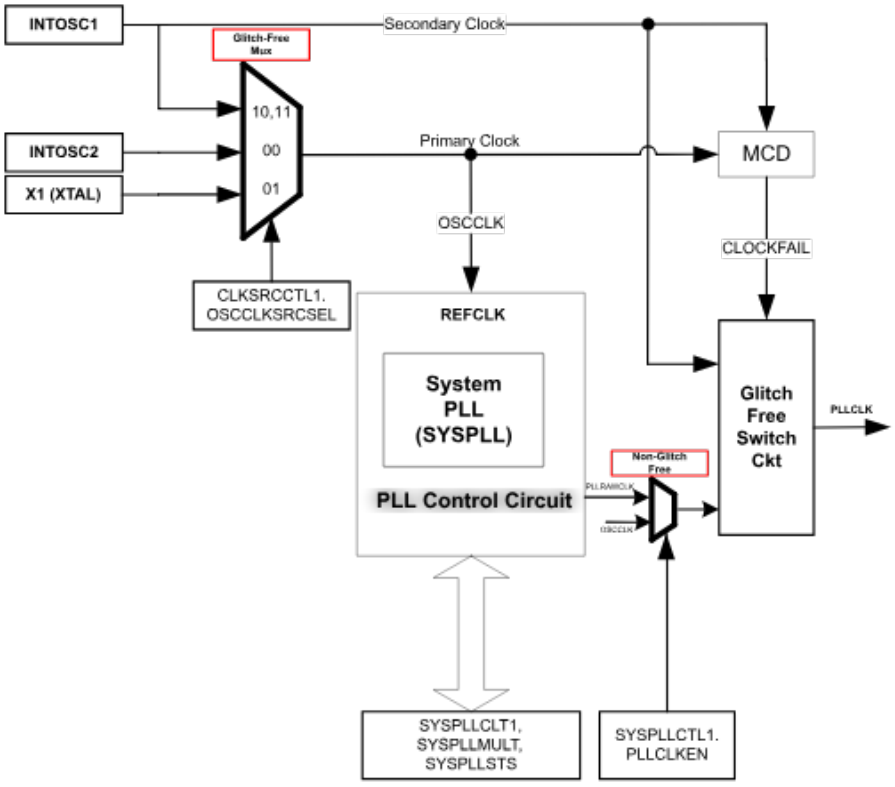


Figure 3-9. Missing Clock Detection Logic

Note

On a complete clock failure when OSCCLK is dead, the operation can take a maximum time of 8192 INTOSC1 cycles (that is, 0.8192ms) before the CLOCKFAIL signal goes high, after which:

- NMI is generated by enabling the CLOCKFAIL signal event configurations in ESM
- OSCCLK is switched to INTOSC1
- PWM Trip happens

3.5.8.2 Dual Clock Comparator (DCC)

DCC can be used for clock quality check as well as clock loss detection. Depending upon the counter settings in DCC, the desired goal can be achieved. In systems where DCC error is indicative of critical failure needing CPU intervention or reset, the DCC error is routed to NMI using ESM. Refer to Chapter 12 for more details on the configuration.

3.6 Bus Architecture

This section covers the details about bus architecture.

3.6.1 Safe Interconnect

Safe interconnect mechanism detects any faults that occur in bus interconnect of C29x CPU and external memory or peripherals. Bus interconnect includes C29x CPU buses (control and data), address decodes, memory controllers and peripheral bridges.

Any data corruption in memory is detected and this is achieved by storing ECC along with data in memory.

To achieve safe interconnect there are checks that are done in the CPU and certain checks are implemented outside CPU. Forward path (CPU to Endpoint) checks are done at endpoint controller. Return path (Endpoint to CPU) checks are done by CPU. This provides end to end safing for the interconnect between initiators (CPU, RTDMA) to memory and peripherals.

In addition to the ECC checks every access placed on the bus has following checks associated :

1. Acknowledge (ACK) : Every access placed on the bus receives ACK from the endpoint (memory controller (in case of memory) or peripheral bridge (in case of peripherals)). This detects any address decode faults that is blocking the access propagation to endpoint.
2. Ready Timeout : Ready timeout checks if for any reason ready is indefinitely pulled low by endpoint due to a fault, timeout logic generates abort request to endpoint. Timeout value is user configurable in SICCONFIG register.

3.6.1.1 Safe Interconnect for Read Operation

CPU generates the ECC for the address and control information needed for the read access to endpoint. This ECC is propagated to memory controller and peripheral bridges along with address and control information.

ECC is for error detection only when the fault is detected the error is sent to ESM via Error Aggregator and CPU goes to fault state. ESM needs to configured to generate NMI and read from address that was captured in error aggregator and user needs to write back the corrected data to fix correctable error at endpoint.

Safe Interconnect mechanisms for read operations :

1. Base Address Check at the Endpoint - Checks for address decode faults
2. ECC Check at the Endpoint - On forward path (CPU to Endpoint) ECC check is performed on address and control information integrity. CPU performs ECC calculation on address and control information locally and send the same to endpoint for cross checking with the ECC calculated at endpoint locally
3. ECC Check at CPU - On return path (Endpoint to CPU) ECC along with Data is fed back to CPU. CPU performs ECC Check

Coverage of the below possible causes of errors are covered by the safe interconnect mechanisms employed above :

1. Address decode fault, for example if the request is routed to the wrong memory due to fault in address decode interconnect logic
2. Address decode execution is correct but on the way to the endpoint if the address bits get flipped resulting in wrong address information sent to endpoint

3.6.1.2 Safe Interconnect for Write Operation

Checks performed are similar to read operation. In addition to address and control the write data is also checked for errors at endpoint in forward path. Write Data ECC generated on combination of address and data is also written along with data to endpoint without any modification if size of access is same as ECC granularity.

In case of endpoints like peripherals that do not support storing of ECC, the ECC is generated at the peripheral bridge so that transactions to and from peripheral bridge are covered.

3.6.2 Peripheral Access Configuration using FRAMESEL

The peripheral access from any initiators (CPUx and RTDMAx) is managed from FRAMES. All peripherals are mapped across 4 FRAME's such that every peripheral instance is accessed at one of the four memory map addresses as shown in Table 3-4 and Table 3-5. The frame selection configuration mechanism allows non-conflicting Initiator to target accesses complete faster and without arbitration stalls. For example, EPWM1 peripheral instance configurations can be accessed from FRAME0 by CPU1 while EPWM2 peripheral instance configurations be accessed from FRAME1 by CPU3. SSU-APR configuration controls the access from CPU to each FRAME region.

Note

Not all peripherals access is managed from FRAMESEL so check the respective Chapter Base Address tables to see if the Structure is Frame Applicable or not.

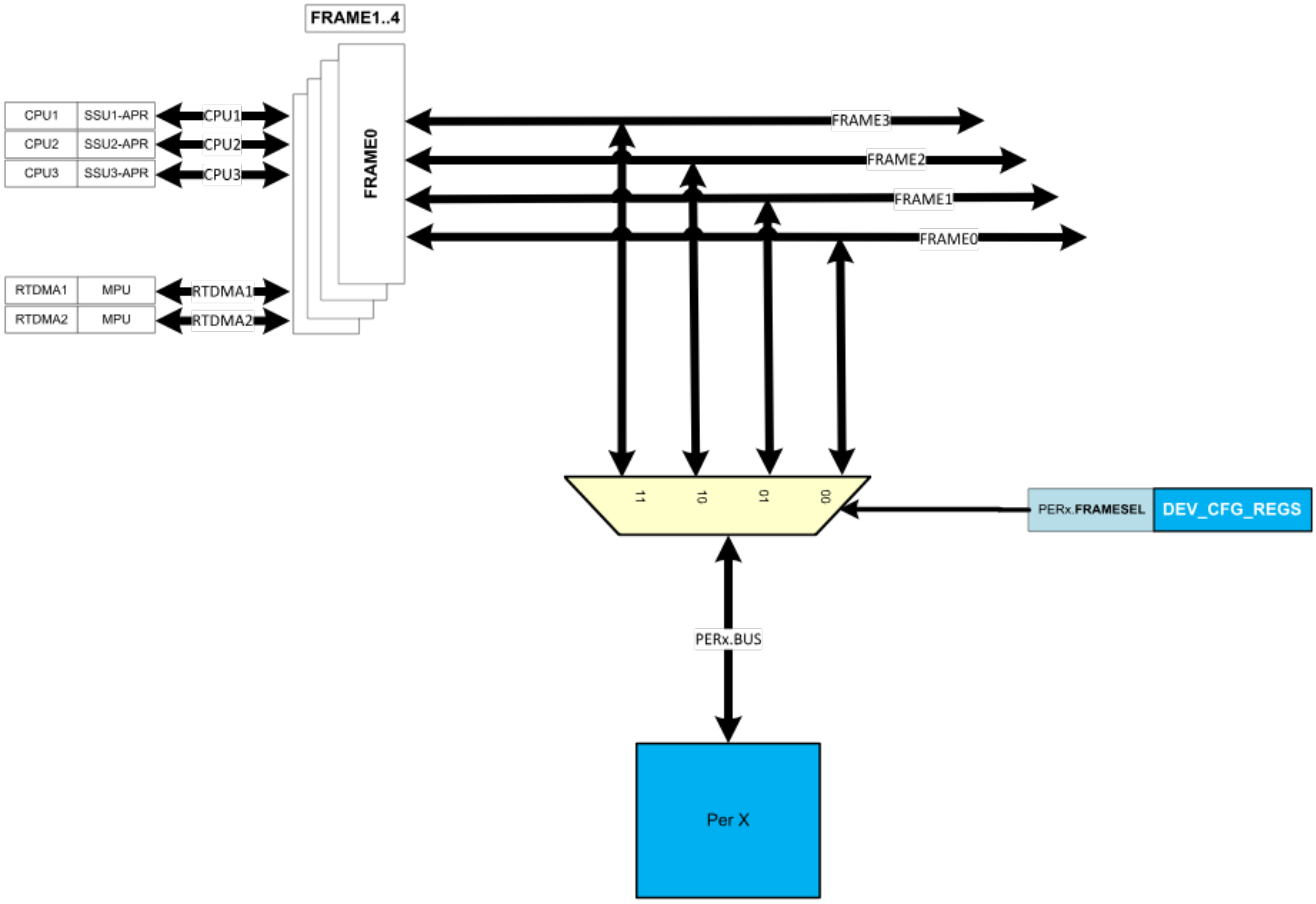


Figure 3-10. Peripheral Access Configuration

Table 3-4. VBUS32 Frame Base Addresses

Peripheral Frame	Base Address
Peripheral Frame 0	7000_0000
Peripheral Frame 1	7040_0000
Peripheral Frame 2	7080_0000
Peripheral Frame 3	70C0_0000

Note

VBUS32 Frame Peripherals do not support x8 access. Any x8 Access to these peripherals gives a Bridge ACK error using Error Aggregator and ESM.

Table 3-5. VBUSP Frame Base Addresses

Peripheral Frame	Base Address
Peripheral Frame 0	6000_0000
Peripheral Frame 1	6040_0000
Peripheral Frame 2	6080_0000
Peripheral Frame 3	60C0_0000

Note

Check data sheet Peripheral Registers Memory Map to see what peripherals are mapped to VBUS32 and VBUSP. All base address tables shown here are FRAME0 addresses. To find the respective FRAME addresses, add the FRAME0 offset to the respective FRAME base addresses.

3.6.3 Bus Arbitration

Access to same peripheral from multiple initiators is handled using the round-robin arbitration. Within round-robin arbitration the priority is given as below :

1. CPU1 Program access
2. CPU2 Program access
3. CPU1 Data accesses
4. CPU2 Data accesses
5. RTDMA1
6. RTDMA2
7. EtherCAT
8. HSM

The arbitration scheme makes sure that no initiator is stalled forever except when intentionally required by user during ATOMIC and burst operations. Back to back access or multiple simultaneous access from same initiator does not block other initiator from getting grant.

Access originating from same initiator is arbitrated using fixed priority as described below in case of CPU :

1. Write
2. Read Bus 1
3. Read Bus 2
4. Program Access

3.7 32-Bit CPU Timers 0/1/2

This section describes the three 32-bit CPU-Timers (TIMER0/1/2) shown in Figure 3-11.

CPU-Timer2 is reserved for real-time operating system uses (for example, TI-RTOS), but if CPU-Timer2 is not used by real-time operating systems then, CPU-Timer2 can also be used for other applications. The CPU-Timer0 and CPU-Timer1 run off of PLLSYSCLK. The CPU-Timer2 normally runs off of PLLSYSCLK, but can also use INTOSC1, INTOSC2 or XTAL. The CPU-Timer interrupt signals (TINT0, TINT1, TINT2) are connected to PIPE Module. Please refer to PIPE Channel mapping table for more details.

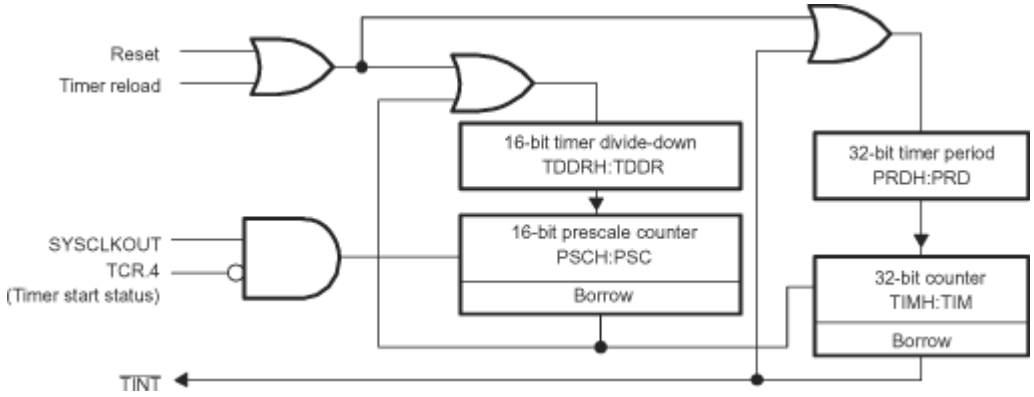


Figure 3-11. CPU Timers

The timer registers are connected to the memory bus of the C29x processor.
The CPU Timers are synchronized to SYSCLKOUT.

The general operation of the CPU-Timer is as follows:

- The 32-bit counter register, TIMH:TIM, is loaded with the value in the period register PRDH:PRD.
- The counter decrements once every (TPR[TDDRH:TDDR]+1) SYSCLKOUT cycles, where TDDRH:TDDR is the timer divider.
- When the counter reaches 0, a timer interrupt output signal generates an interrupt pulse.

The registers listed in Section 3.13 are used to configure the timers.

3.8 Watchdog Timers

The watchdog module generates an output pulse 512 watchdog-clocks (WDCLKs) wide whenever the 8-bit watchdog up counter has reached the maximum value. The watchdog clock source is INTOSC1. Software must periodically write a 0x55 + 0xAA sequence into the watchdog key register to reset the watchdog counter. The counter can also be disabled. Figure 3-12 shows the various functional blocks within the watchdog module.

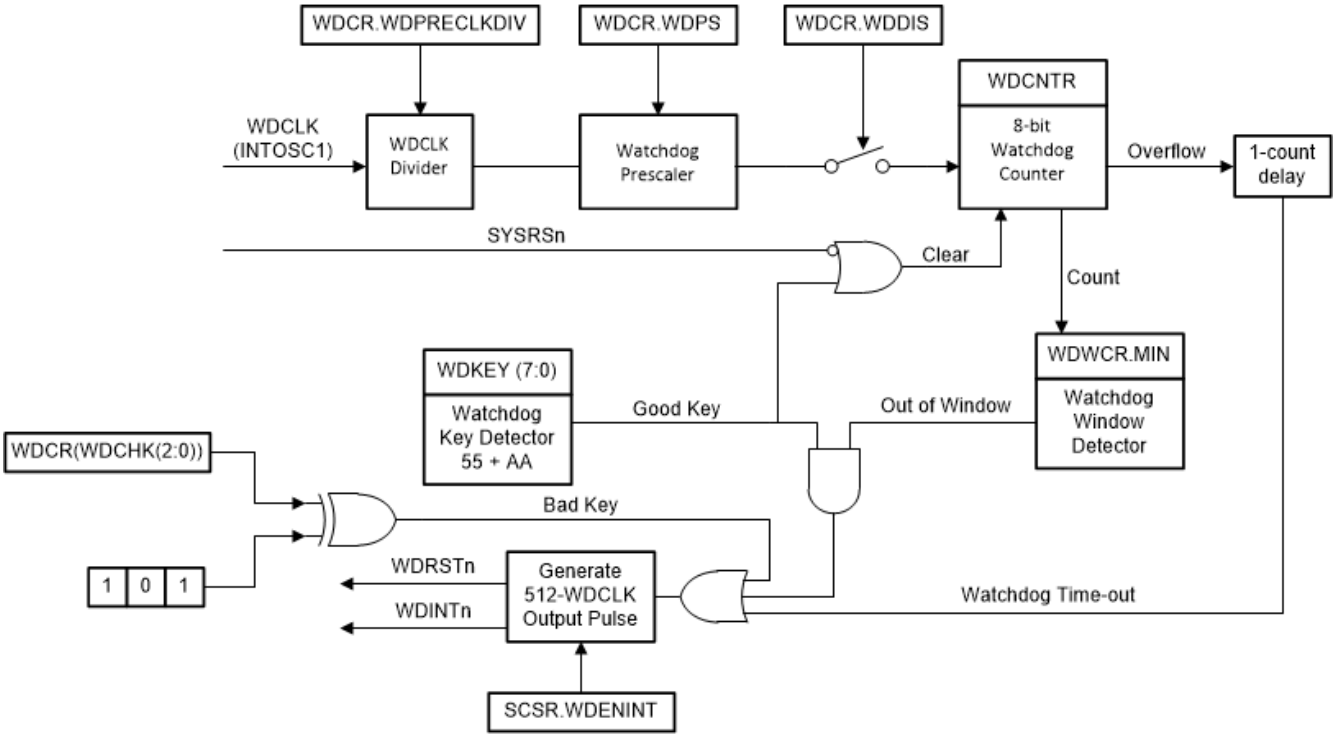


Figure 3-12. CPU Watchdog Timer Module

3.8.1 Servicing the Watchdog Timer

The watchdog counter (WDCNTR) is reset when the proper sequence is written to the WDKEY register before the 8-bit watchdog counter overflows. The WDCNTR is reset-enabled when a value of 0x55 is written to the WDKEY. When the next value written to the WDKEY register is 0xAA, then the WDCNTR is reset. Any value written to the WDKEY other than 0x55 or 0xAA causes no action. Any sequence of 0x55 and 0xAA values can be written to the WDKEY without causing a system reset; only a write of 0x55 followed by a write of 0xAA to the WDKEY resets the WDCNTR.

Table 3-6. Example Watchdog Key Sequences

Step	Value Written to WDKEY	Result
1	0xAA	No action
2	0xAA	No action
3	0x55	WDCNTR is enabled to be reset if next value is 0xAA.
4	0x55	WDCNTR is enabled to be reset if next value is 0xAA.
5	0x55	WDCNTR is enabled to be reset if next value is 0xAA.
6	0xAA	WDCNTR is reset.
7	0xAA	No action
8	0x55	WDCNTR is enabled to be reset if next value is 0xAA.
9	0xAA	WDCNTR is reset.
10	0x55	WDCNTR is enabled to be reset if next value is 0xAA.
11	0x32	Improper value written to WDKEY. No action, WDCNTR no longer enabled to be reset by next 0xAA.
12	0xAA	No action due to previous invalid value.
13	0x55	WDCNTR is enabled to be reset if next value is 0xAA.
14	0xAA	WDCNTR is reset.

Step 3 in Table 3-6 is the first action that enables the WDCNTR to be reset. The WDCNTR is not actually reset until step 6. Step 8 again re-enables the WDCNTR to be reset and step 9 resets the WDCNTR. Step 10 again re-enables the WDCNTR to be reset. Writing the wrong key value to the WDKEY in step 11 causes no action, however the WDCNTR is no longer enabled to be reset and the 0xAA in step 12 now has no effect.

If the watchdog is configured to reset the device, then a WDCNTR overflow or writing the incorrect value to the WDCR[WDCHK] bits resets the device and sets the watchdog flag (WDRSn) in the reset cause register (RESC). After a reset, the program can read the state of this flag to determine whether the reset was caused by the watchdog. After doing this, the program must clear WDRSn to allow subsequent watchdog resets to be detected. Watchdog resets are not prevented when the flag is set.

3.8.2 Minimum Window Check

To complement the timeout mechanism, the watchdog also contains an optional "windowing" feature that requires a minimum delay between counter resets. This can help protect against error conditions that bypass large parts of the normal program flow but still include watchdog handling.

To set the window minimum, write the desired minimum watchdog count to the WDWCR register. This value takes effect after the next WDKEY sequence. From then on, any attempt to service the watchdog when WDCNTR is less than WDWCR triggers a watchdog interrupt or reset. When WDCNTR is greater than or equal to WDWCR, the watchdog can be serviced normally.

At reset, the window minimum is zero, which disables the windowing feature.

3.8.3 Watchdog Reset or Watchdog Interrupt Mode

The watchdog can be configured in the SCSR register to either reset the device ($\overline{\text{WDRST}}$) or assert an interrupt ($\overline{\text{WDINT}}$), if the watchdog counter reaches the maximum value. The behavior of each condition is:

- **Reset mode:** If the watchdog is configured to reset the device, then the $\overline{\text{WDRST}}$ signal pulls the device reset ($\overline{\text{XRS}}$) pin low for 512 INTOSC1 cycles when the watchdog counter reaches the maximum value.
- **Interrupt mode:** When the watchdog counter expires, the watchdog asserts an interrupt by driving the $\overline{\text{WDINT}}$ signal low for 512 INTOSC1 cycles. $\overline{\text{WDINT}}$ is mapped to the PIPE module to assert appropriate CPU interrupt line as configured.

  An error event input is also generated to ESM (for more details refer to ESM Inputs section in ESM Chapter) which can be configured to take actions as deemed necessary by system integrator.

  To avoid unexpected behavior, software must not change the configuration of the watchdog while $\overline{\text{WDINT}}$ is active. For example, changing from interrupt mode to reset mode while $\overline{\text{WDINT}}$ is active immediately resets the device. Disabling the watchdog while $\overline{\text{WDINT}}$ is active causes a duplicate interrupt, if the watchdog is later re-enabled. If a debug reset is issued while $\overline{\text{WDINT}}$ is active, the reset cause register (RESC) shows a watchdog reset. The WDINTS bit in the SCSR register can be read to determine the current state of $\overline{\text{WDINT}}$.

3.8.4 Watchdog Operation in Low-Power Modes

In IDLE mode, the watchdog interrupt ($\overline{\text{WDINT}}$) signal can generate an interrupt to the CPU to take the CPU out of IDLE mode. User software must determine which peripheral caused the interrupt.

In STANDBY mode, clocks to the peripherals are turned off based on LPMCR, PCLKCRx and STANDBYEN registers. The CPU clock is gated when entering STANDBY mode based on LPMCR.LPM configuration but the watchdog remains functional since the module runs off the internal oscillator clock (INTOSC1). The $\overline{\text{WDINT}}$signal is applied to the Low Power Modes (LPM) block, so that the LPM block can be used to wake the CPU from STANDBY low-power mode. This feature is enabled by setting LPMCR.WDINTE = 1. See Section 3.9 for details. CPU can be brought out of IDLE or STANDBY mode using any of the enabled INT, RINT or NMI.

Note

If the watchdog interrupt is used to wake-up from an IDLE or STANDBY low-power mode condition, software must make sure that the $\overline{\text{WDINT}}$ signal goes back high before attempting to reenter the IDLE or STANDBY mode. The $\overline{\text{WDINT}}$ signal is held low for 512 INTOSC1 cycles when the watchdog interrupt is generated. The current state of $\overline{\text{WDINT}}$ can be determined by reading the watchdog interrupt status (WDINTS) bit in the SCSR register. WDINTS follows the state of $\overline{\text{WDINT}}$ by two SYSCLKOUT cycles.

3.8.5 Emulation Considerations

The watchdog module behaves as follows under various debug conditions:

CPU Suspended:	When the CPU is suspended, the watchdog clock (WDCLK) is suspended
Run-Free Mode:	When the CPU is placed in run-free mode, then the watchdog module resumes operation as normal.
Real-Time Single-Step Mode:	When the CPU is in real-time single-step mode, the watchdog clock (WDCLK) is suspended. The watchdog remains suspended even within real-time interrupts.
Real-Time Run-Free Mode:	When the CPU is in real-time run-free mode, the watchdog operates as normal.

3.9 Low-Power Modes

This device has two clock-gating, low-power modes. All low-power modes are entered by setting the LPMCR register and executing the IDLE instruction. More information about this instruction can be found in the *F29x CPU Reference Guide*.

Low-power modes cannot be entered into while a Flash program or erase is ongoing. The application can verify the following before entering STANDBY:

1. Check the value of the GPIODAT register of the pin selected for STANDBY (GPIOLPMSEL0/1) prior to entering the low-power mode to make sure that the wake event has not already been asserted.
2. The LPMCR.QUALSTDBY register can be set to a value greater than the ratio of INTOSC1/PLLSYSCLK to make sure proper wake up.

3.9.1 IDLE

IDLE is a standard feature of the CPU. In this mode, the CPU is turned off while CPUx.CLOCK and all peripheral clocks are left running. IDLE can thus be used to conserve power while a CPU is waiting for peripheral events. When one CPU is in IDLE, there is no effect on the other CPU subsystem.

Any enabled interrupt wakes up the CPU from IDLE mode.

To enter IDLE mode, set LPMCR.LPM to 0x0 and execute the IDLE instruction.

3.9.2 STANDBY

STANDBY is a more aggressive low-power mode that gates both the CPU clock and any peripheral clocks derived from PLLSYSCLK (based on STANDBYEN setting). The watchdog however, is left active. Like IDLE, this mode affects only one CPU subsystem. The other CPU subsystem and all of the peripherals are unaffected. STANDBY is good for an application where the wake-up signal comes from an external system (or CPU subsystem) rather than a peripheral input.

An NMI (or optionally) any interrupt or a configured GPIO can wake the CPU from STANDBY mode. Each GPIO from GPIO0-63 can be configured to wake the CPU when the GPIO are driven active low. Additionally a CMPSS based wake up can be used for coming out of standby mode by configuring CMPSSLPMSEL register.

IPC interrupt 1 (flag 0), an NMI fired to the other CPU, or (optionally) any interrupt, wakes the CPU subsystem up from STANDBY mode. Any of GPIO0-63 can also be configured to wake up the subsystem when the GPIO is driven active low.

To enter STANDBY mode:

1. Set LPMCR.LPM to 0x1.
2. Enable the required interrupt in the PIPE
3. For watchdog interrupt wakeup, set LPMCR.WDINTE to 1 and configure the watchdog to generate interrupts.
4. For GPIO wakeup, set GPIOLPMSEL0 and GPIOLPMSEL1 to connect the chosen GPIOs to the LPM module, and set LPMCR.QUALSTDBY to select the number of OSCCLK cycles for input qualification.
5. Execute the IDLE instruction to enter STANDBY.

To wake up from Standby mode:

1. Configure the desired GPIO to trigger the wakeup.
2. Drive the selected GPIO signal low; the signal must remain low for the number of OSCCLK cycles specified in the QUALSTDBY bits in the LPMCR register. If the signal is sampled high during this period, the count restarts.

At the end of the qualification period, the PLL enables the CLKIN to the CPU and the associated wake up interrupt is latched in the PIPE module.

The CPU is now out of STANDBY mode and can resume normal execution.

3.10 Memory Subsystem (MEMSS)

This section discusses the memory subsystem of the device and how the buses, arbitration, and memory controllers are configured.

3.10.1 Introduction

The MEMSS, or Memory Subsystem, covers the memory architecture (RAM and ROM) used on the C29x platform. There are multiple initiators accessing the memories on the device like C29 CPU's, RTDMA's, HSM and EtherCAT.

Each CPU has a 128-bit program bus, two 64-bit read buses, and a 64-bit write bus. Hence, there are separate 128-bit and 64-bit memory controllers that are optimized based on program or data access from specific CPU pairs.

Certain memories are optimized for zero wait state access from specific CPU pair (CPU1/2 or CPU1/3) and type of access (program or data) as described in Table 3-7. Access from other initiators like RTDMA is pipelined but burst mechanism is added between RTDMA and MEMSS to improve throughput.

The MEMSS registers can be found in Section 3.13.

Table 3-7. Naming Conventions

Name	Read Word Access	Zero Wait State Optimization
LPAx RAM	128-bit word	Program Access for CPU1 and CPU2
LDAx RAM	64-bit word	Data Access for CPU1 and CPU2
CPAx RAM	128-bit word	Program Access for CPU1 and CPU3
CDAx RAM	64-bit word	Data Access for CPU1 and CPU3

3.10.2 Features

The MEMSS implements the following features for memory:

- RAM:
  - RTDMA throughput optimization with local lookahead address generation
  - Common dataline buffer for each CPU (2x64-bit words)
  - Common program bridge for each CPU
  - ECC support with 32-bit granularity
  - Read-modify-write for write access smaller than ECC granularity
  - Posted write to minimize stalls on read-modify-write operation
  - Test mode to read/write ECC bits and error injection
- ROM:
  - ECC support with 64-bit granularity to reduce ECC bits overhead
  - One wait state program and data access
  - Prefetch with 256-bit wide memory
  - Dedicated local line buffer of 256 bits
- To reduce ECC bit overhead, there are no separate address ECC bits; ECC is generated by combining data and address

3.10.3 Configuration Bits

Each memory block has the following configuration bits:

- Test mode to read ECC bits and error injection (TESTMODE)
- Memory initialization trigger bit (MEMINIT)

These bits can be configured by CPU1 LINK0, LINK1, or LINK2. Debug accesses are only allowed if Zone 0 or Zone 1 are enabled for full-debug by all CPUs.

3.10.3.1 Memory Initialization

Memory initialization is initiated by writing to a software-programmable bit. Each memory has independent memory initialization control. Accesses are stalled for the memory being initialized until the initialization is complete. MEMINIT can be set only from CPU1 LINK0, LINK1, or LINK2.

3.10.4 RAM

This section details the architecture and configuration of the MEMSS for RAM.

3.10.4.1 MEMSS Architecture

Table 3-8 shows the different wait states for accessing different memory sections, depending on the initiator that is accessing the memory.

Note

In Table 3-8, nWS indicates a wait state of n cycles.

Table 3-8. Memory Configuration

RAM Section	Interleaved	CPU1	CPU2	CPU3	HSM	RTDMA1	RTDMA2
LPAx RAM	Yes	0WS program 1WS data	0WS program 1WS data	3WS data		1WS	1WS
LDAx RAM	Yes	1WS program 0WS data	1WS program 0WS data	3WS data	2WS	1WS	1WS
M0 RAM	Yes	0WS data	0WS data (read-only)	3WS data (read-only)			
CPAx RAM	Yes	0WS program 1WS data	3WS data	0WS program 1WS data		1WS	1WS
CDAx RAM	Yes	1WS program 0WS data	3WS data	1WS program 0WS data		1WS	1WS
CPU1 ROM	Yes	1WS program 1WS data					
CPU2 ROM	Yes		1WS program 1WS data				
CPU3 ROM	Yes			1WS program 1WS data			

Figure 3-13 shows the integration view of all the RAM and the respective accesses.

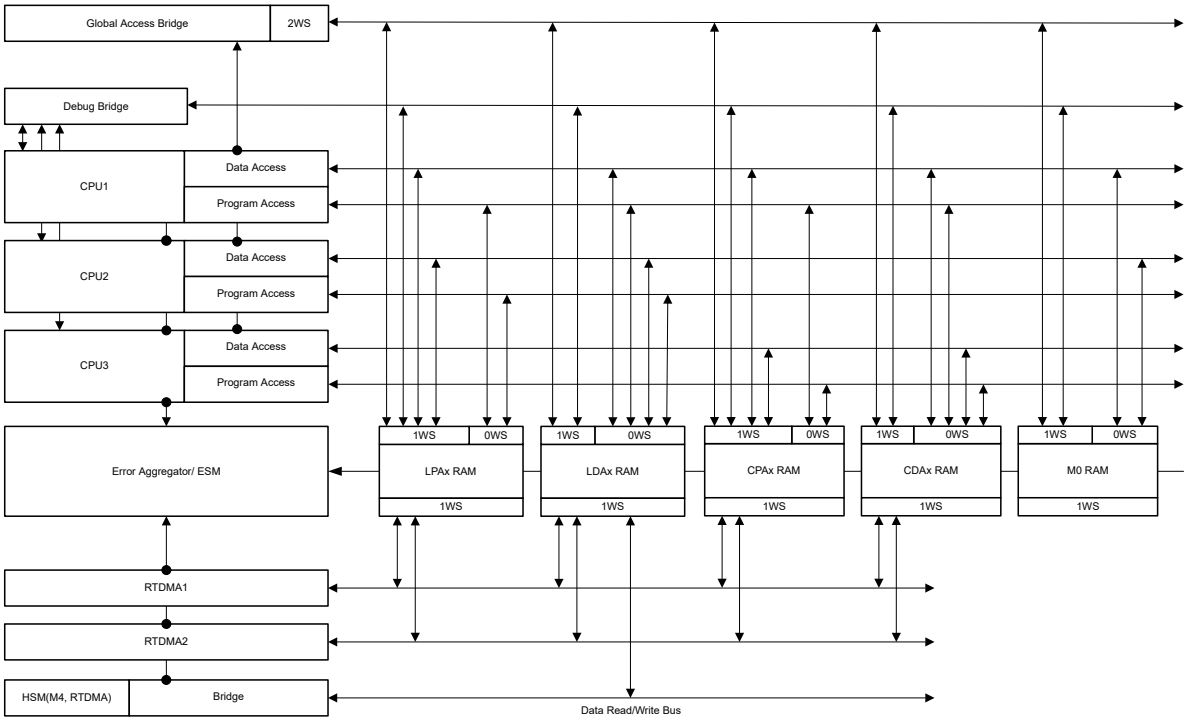


Figure 3-13. Integration View of All RAMs

3.10.4.2 RAM Memory Controller Overview

Each CPU has a dataline buffer in common for all RAMs. This buffer is only used for RAM accesses, and accesses to other regions of the memory map are not be buffered or served from the dataline buffer. The Flash read interface and ROM controllers have dedicated line buffers local to the controller and are 256-bit wide memories.

Each CPU has a program bridge to interface 64-bit memories. This bridge is common for all 64-bit RAMs connected to a CPU and bridges the 128-bit bus of the CPU and the 64-bit buses of the LDx and CDx memory controller. Global bridge access allows data access to any RAM from any CPU which does not already have direct access, with a tradeoff of being three wait states to access the memory.

Memory controllers have a fast access port and slow access port. Fast access ports are zero wait state and limited, while slow access ports are one wait state and can have improved throughput if the initiator supports burst mode. Each CPU also has a debug access port, which are arbitrated externally. Only one port is routed to the memory controller. Debug accesses are connected to slow access ports.

Safe interconnect checks are done in initiators and memory controllers. Any errors resulting from HSM accesses are aggregated in the HSM-ESM, whiles others are aggregated in the ESM.

ECC granularity for RAM data is 32 bits. Write accesses of 8 or 16 bits require a read-modify-write operation. M0 memory is the similar to LDAx memory, and access permissions are determined by the SSU.

The LDAx memories are accessible from the CPU and HSM and can be used as shared memories or dedicated memories. Freedom from interference is provided so the CPU does not get access to secure HSM data. The LDAx memories have a multi-layer protection:

- HSM controls shared memory access using independent USURP control for each LDAx memory block (enables/blocks access completely)
- CPU enables the access to LDAx memories using MPU present in HSM bridge. Only access to selected portion is allowed
- Two additional levels of protection that can be enabled with the SSU and other MPUs in the system

Read and write to the LDAx memories is only allowed if both systems allow the access type, otherwise an access violation error is generated to the respective system.

3.10.4.3 Memory Controllers

3.10.4.3.1 128-Bit LPx and CPx Memory Controller

The 128-bit memory controller enables zero wait states on program accesses. This also allows data accesses, providing the ability to copy code, download code, and insert software breakpoints. Data accesses can be one or more wait states. Each logical block of the 128-bit memory has four memory banks. The data width of these RAMs is 39 bits (32-bit data and 7-bit ECC). The four banks are grouped together and the addresses are interleaved to form a 128-bit word. All data reads are treated as 64-bit reads, and 64-bit data is returned because the dataline buffer is 64-bit wide.

The 128-bit memory controllers have accesses from the following initiators:

- CPU1,2,3
- RTDMA1-2
- Debug

Figure 3-14 shows a view of the 128-bit memory.

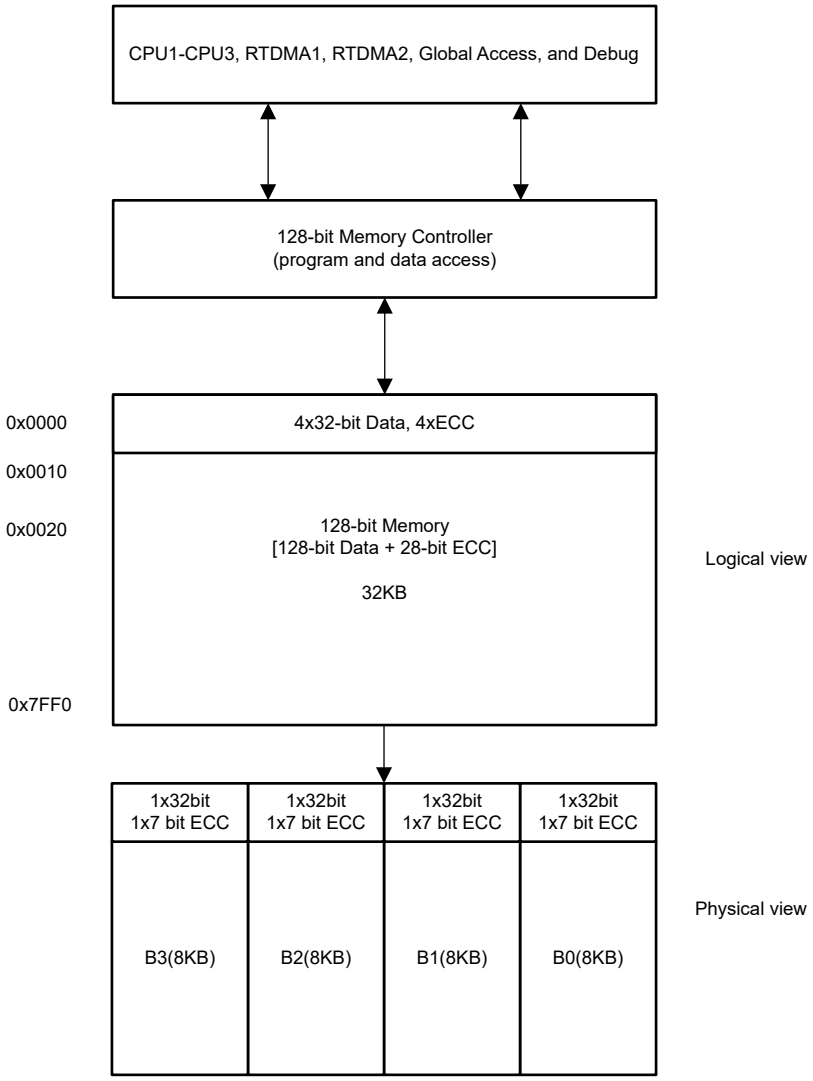


Figure 3-14. 128-Bit Memory

Figure 3-15 shows the 128-bit memory controller. This controller implements LPx and CPx memories. Each instance of the controller is optimized for zero wait states from two specific CPUs (CPUa and CPUb). For example, LPAx is optimized for CPU1 and CPU2 program access while CPAx is optimized for CPU1 and CPU3 program access. For these configurations, the controller is the same but the CPUs that get connected as CPUa and CPUb are different.

Note

Program accesses from CPUs other than CPUa and CPUb are not supported. For example, LPAx does not have CPU3 program access.

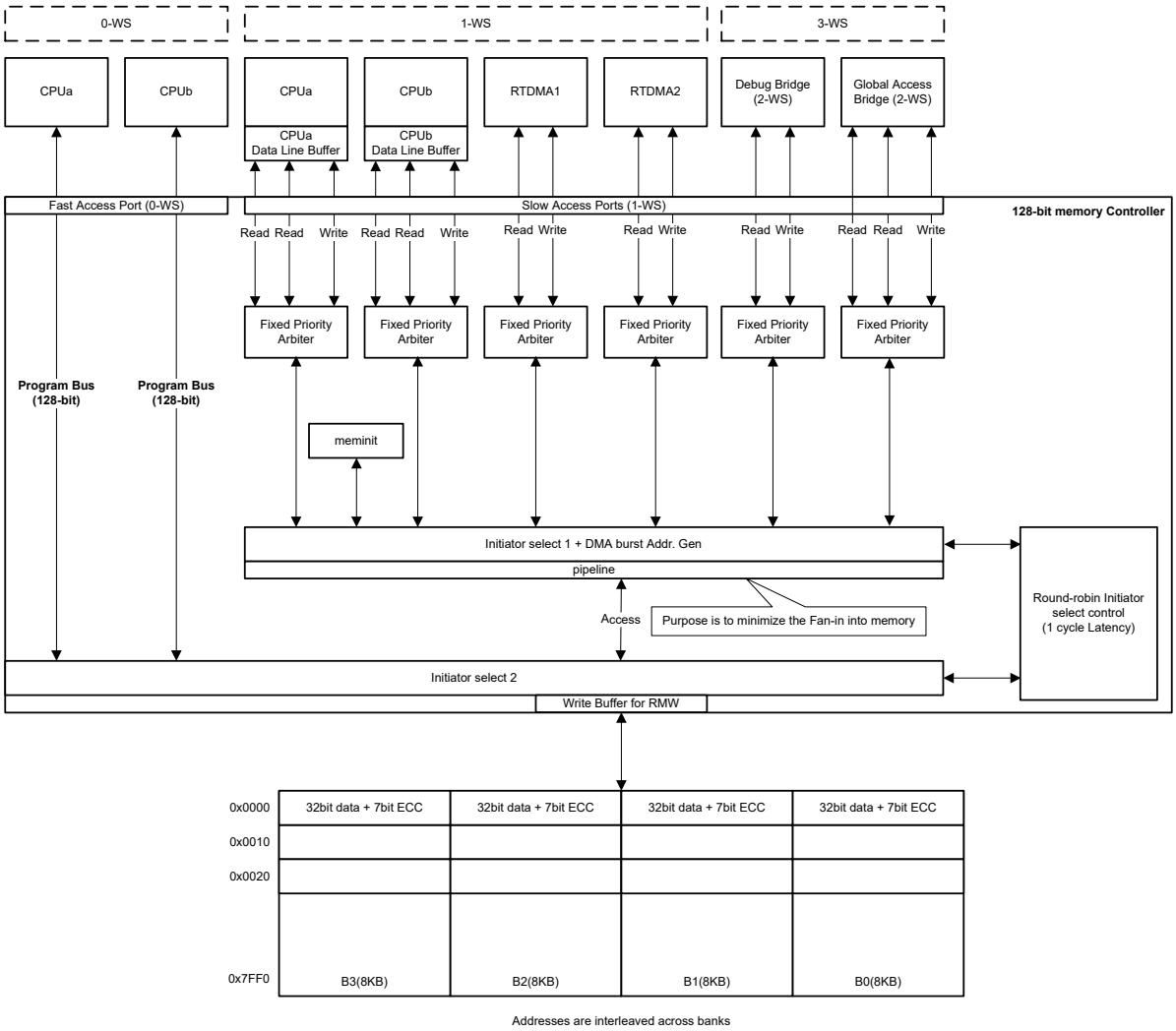


Figure 3-15. 128-Bit Memory Controller

The 128-bit controller has fast and slow access ports. Fast access ports have zero wait state program access from CPUa and CPUb. Slow access ports are intended for non-critical data access and have one wait state due to pipelining. Lookahead reads using the RTDMA burst signaling is supported on slow access ports.

Note

CPUa and CPUb use the respective clock domain when accessing memory.

The read-modify write operation happens in the background without stalling the CPU, using the write buffer close to the memory.

Data accesses from CPUs other than CPUa and CPUb are routed through global access bridge. Debug access is routed through debug access bridge. The latency of these accesses is three wait states. Two wait states are introduced by the global access bridge for data access and debug access bridge for debug access with additional one wait state introduced by the slow access port of 128-bit memory controller.

Note

There is no restriction on using a portion of the program RAM as data memory, so long as the SSU setting allows for this. However, this is not recommended as arbitration affects CPU performance.

The following is the round-robin priority order:

1. CPUa Program Access
2. CPUb Program Access
3. CPUa Data Access
4. CPUb Data Access
5. RTDMA1
6. RTDMA2
7. Debug Access Bridge
8. Global Access Bridge

3.10.4.3.2 64-Bit LDx and CDx Memory Controller

The 64-bit LDx and CDx memory controller is similar to the 128-bit LPx and CPx memory controller, but differs in a few ways. The 64-bit memory controller has zero wait states on data access and one wait state on program access. Each block of 64-bit memory has two memory banks. The data width of these RAMs is 39 bits (32-bit data and 7-bit ECC). The two banks are grouped together and the addresses are interleaved to form a 64-bit word.

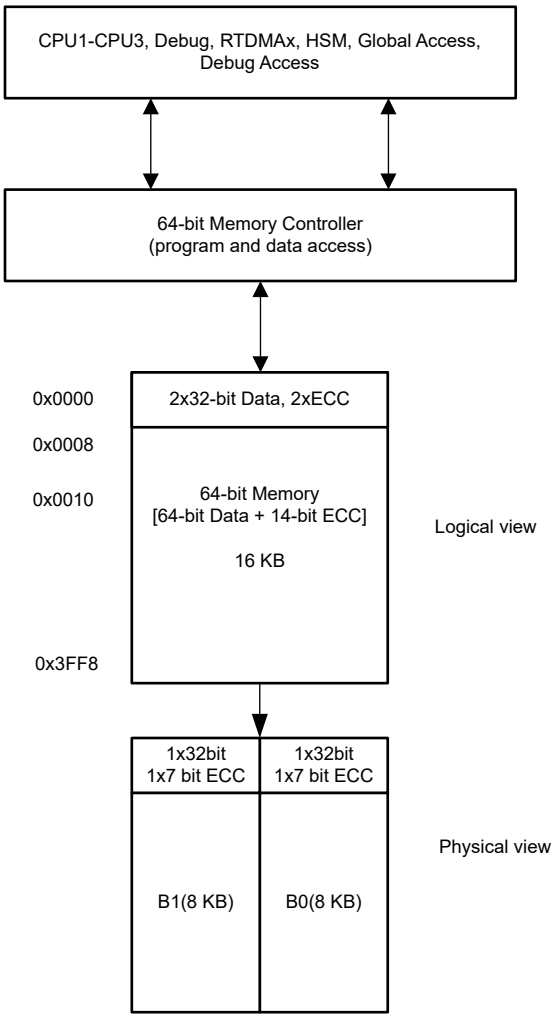


Figure 3-16. 64-Bit Memory

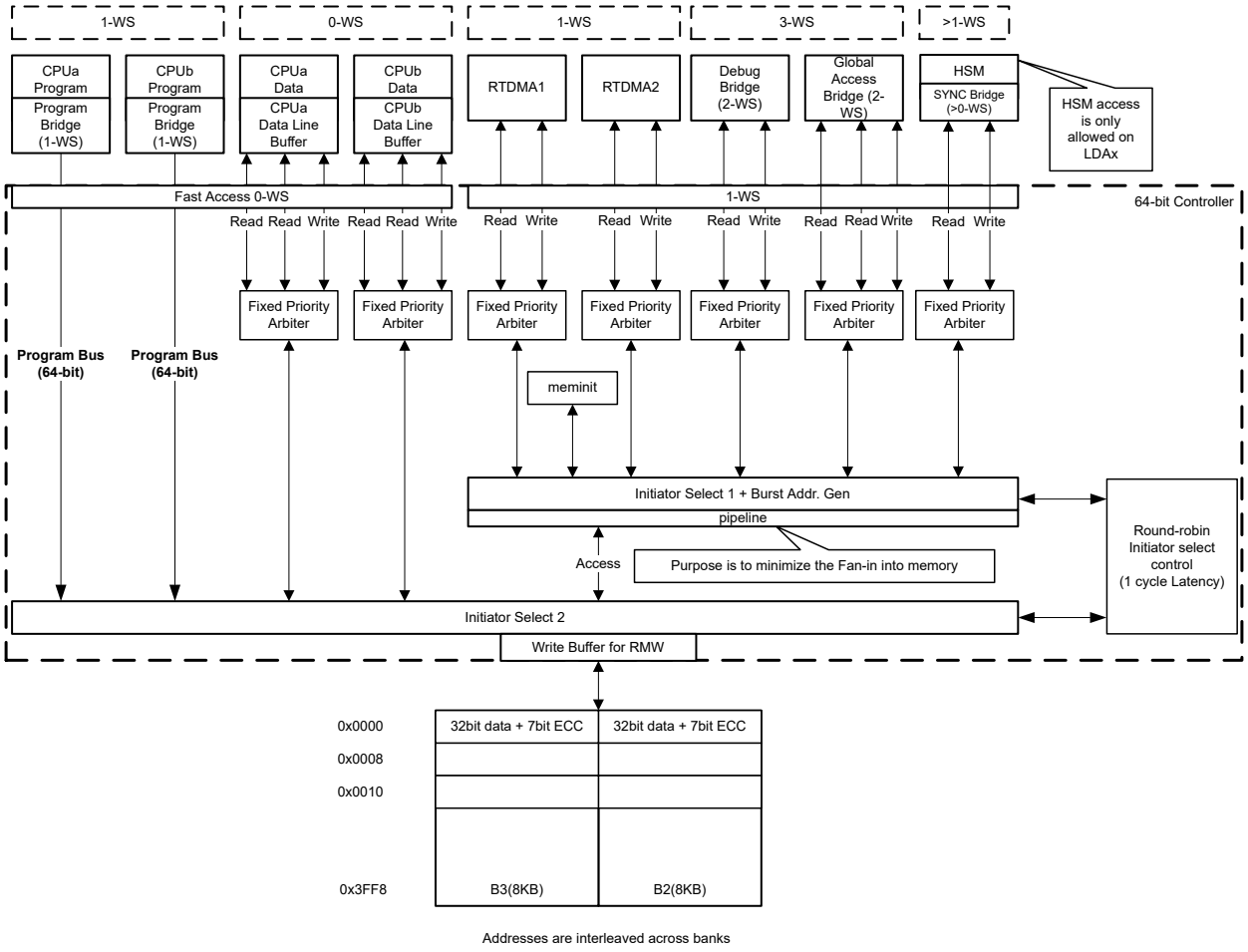


Figure 3-17. 64-Bit Memory Controller

Fast access ports have zero wait states for program and data access from CPUa and CPUb. For example, LDAx RAM has fast access port access for CPU1 and CPU2 program and data access while CDAx RAM has fast port access for CPU1 and CPU3 program and data access.

Program access from other CPUs is not supported. For example, CPU3 does not have program access for LDAx RAM. Fast access ports support 64-bit program access, and the program access bridge connects the 64-bit memory interface to the 128-bit memory bus of the CPU. The accesses between the program access bridge and the 64-bit memory controller are zero wait states, but the bridge initiates two accesses to form a 128-bit word. This is effectively one wait state for a 128-bit word with respect to CPU. The CPUa and CPUb data write access ports have one level of buffering, and the write buffer also handles read-modify-write operations.

The following is the round-robin priority order:

1. CPUa Program Access
2. CPUb Program Access
3. CPUa Data Access
4. CPUb Data Access
5. RTDMA1
6. RTDMA2

7. HSM
8. Debug Access Bridge
9. Global Access Bridge

Other details regarding the 64-bit memory controller are the same as the 128-bit memory controller, with the only difference being the width of the memory controller. Refer to Section 3.10.4.3.1 for more information.

3.10.4.3.3 M0 Memory Controller

M0 memory is similar to LDAx memory, but M0 only has accesses from the CPUs. Below are some restrictions on M0. Refer to SSU Chapter for more details.

- No RTDMA access
- Only CPU1 has permission read/write
- Only read is supported from CPU2 and CPU3
- Fetch is not allowed

3.10.4.4 RTDMA Burst Support

RTDMA is connected through a slow access port as shown in Figure 3-18, so all accesses are minimum one wait state. To improve throughput, RTDMA supports local address generation within the MEMSS memory controller. This enables performance close to zero wait states.

Local address generation within a burst is enabled by the following signals:

- First access of burst sequence (FIRST)
- Last access of burst sequence (LAST)
- Address Increment for next address (XCNT)
  - Next Address = Current Address + XCNT; this is for lookahead access generation

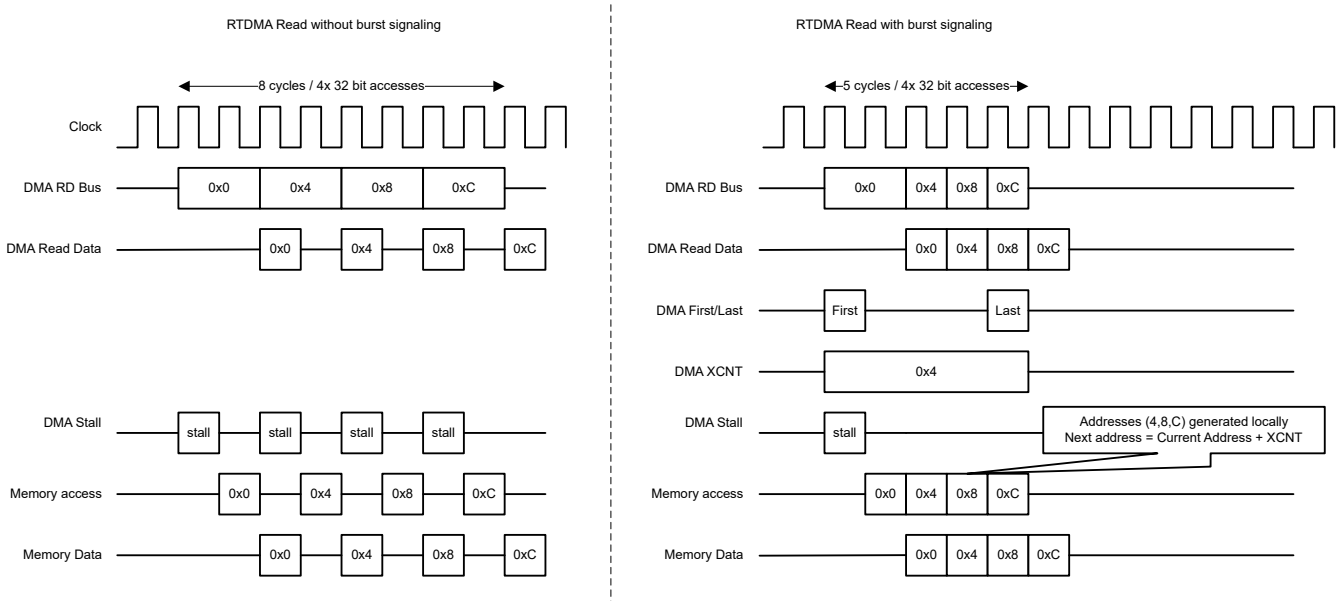


Figure 3-18. RTDMA Read

Without burst signaling, the address is latched first and then presented to memory. With burst, this happens only once for the first access, since the MEMSS memory controller does not use addresses generated by the RTDMA for subsequent accesses, as addresses is generated locally.

The LAST signal is an indication to stop the lookahead read. If the LAST signal is not generated, the RTDMA request going inactive is treated as a burst termination.

The RTDMA arbitration is locked when the FIRST access of a burst is granted, and the arbitration is released on the LAST access. Accesses with the FIRST and LAST signals asserted are treated as non-burst, and a lookahead read is not initiated nor is the arbitration locked. The arbitration lock is removed if the RTDMA channel is halted due to a CPU halt.

3.10.4.5 Atomic Memory Operations

The ATOMIC.M, or atomic, operation executes a protected sequence of memory operations while preventing other memory initiators from interrupting. This sequence can involve reading and updating shared variables in memory, where there is a need to protect these variables from updates by other initiators. The CPU atomic operation feature is typically used to implement a mutex or semaphore operation on shared data.

For true atomic CPU operations, the atomic operation must extend to memory accesses. The arbitration is locked to the CPU doing the atomic operations as soon as access is granted to the CPU. This can be a read or write operation. The lock is released when both signals transition low. From this point, other initiators can access the memory. This is visualized in Figure 3-19.

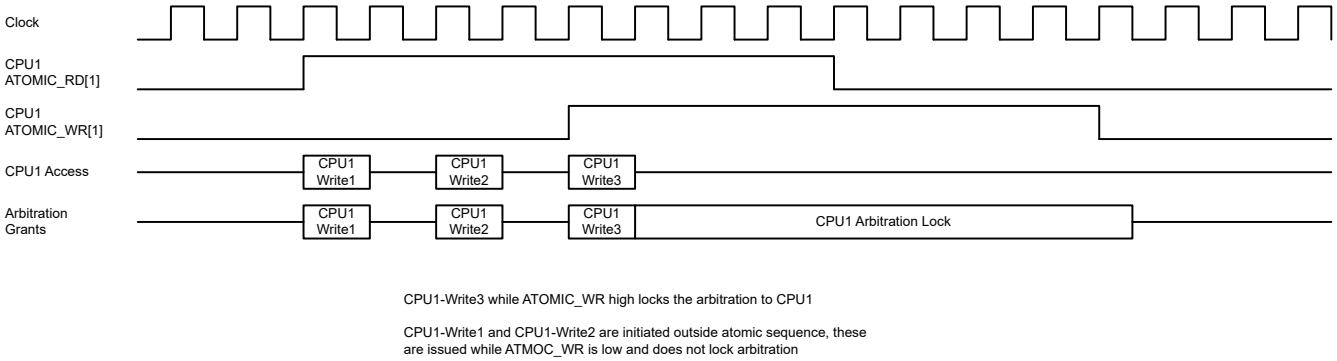


Figure 3-19. Atomic Operations in the MEMSS

The CPU atomic operation extends to a maximum of 256 instruction packets. For atomic operations, reads are always directed to the memory controller to fetch the latest data from memory to make sure the latest data is being returned.

The arbitration lock is removed if the corresponding CPU enters a halt state. This prevents other initiators from being stalled forever and causing the CPU timeout checkers to generate a fault in the case where a CPU is halted. For this reason, atomic operations among initiators is not specified under debug scenarios. To maintain an atomic sequence during debug the DBGM control must be used to mark the atomic sequence as non-haltable.

Read-modify-write and memory-to-memory data move CPU instruction types are treated as regular reads and writes by the MEMSS. To have these operations treated as a protected sequence, the instructions need to be explicitly included in an atomic section of the code.

More than one memory controller can be locked to a CPU if the atomic block of code accesses more than one memory block. The lock is released simultaneously for MEMSS memory controllers at the end of an atomic sequence. If there are back-to-back atomic blocks being executed on a CPU, the MEMSS memory controller can be locked to that CPU through all blocks of code.

If a CPU has both an atomic read and atomic write, the access arbitration follows the fixed priority scheme. If two or more CPUs have atomic accesses, the arbitration is locked to one CPU and moves round-robin among the other CPUs. When some CPUs have non-atomic accesses, the accesses are serviced in between each access in the round-robin scheme. For example, if CPU1 has an atomic access, CPU2 has a non-atomic access, and CPU3 has an atomic access, and CPU1 is currently being serviced with an atomic lock, then one access for CPU2 is serviced followed by an atomic lock for CPU3.

3.10.4.6 RAM ECC

All RAMs have ECC protection with a 32-bit granularity. This means that every 32 bits has 7 bits of ECC. ECC for write data is generated in the CPU and written to memory directly. The read-modify-write operation is performed within the MEMSS memory controller for accesses that are less than 32 bits, and the ECC is generated locally for these write accesses.

Note

There is no separate address ECC. The ECC code generated is a combination of the address and data.

3.10.4.7 Read-Modify-Write Operations

Memory initiators are byte addressable and can generate write accesses that are 8/16/32/64 bits. A read-modify-write operation is performed locally in the MEMSS memory controller for 8 and 16-bit write accesses.

3.10.4.8 Dataline Buffer

Figure 3-20 shows a high level view of the dataline buffer scheme. LB1 and LB2 are two data Line Buffers. The buffer is reset by the respective CPUx.SYSRSn reset.

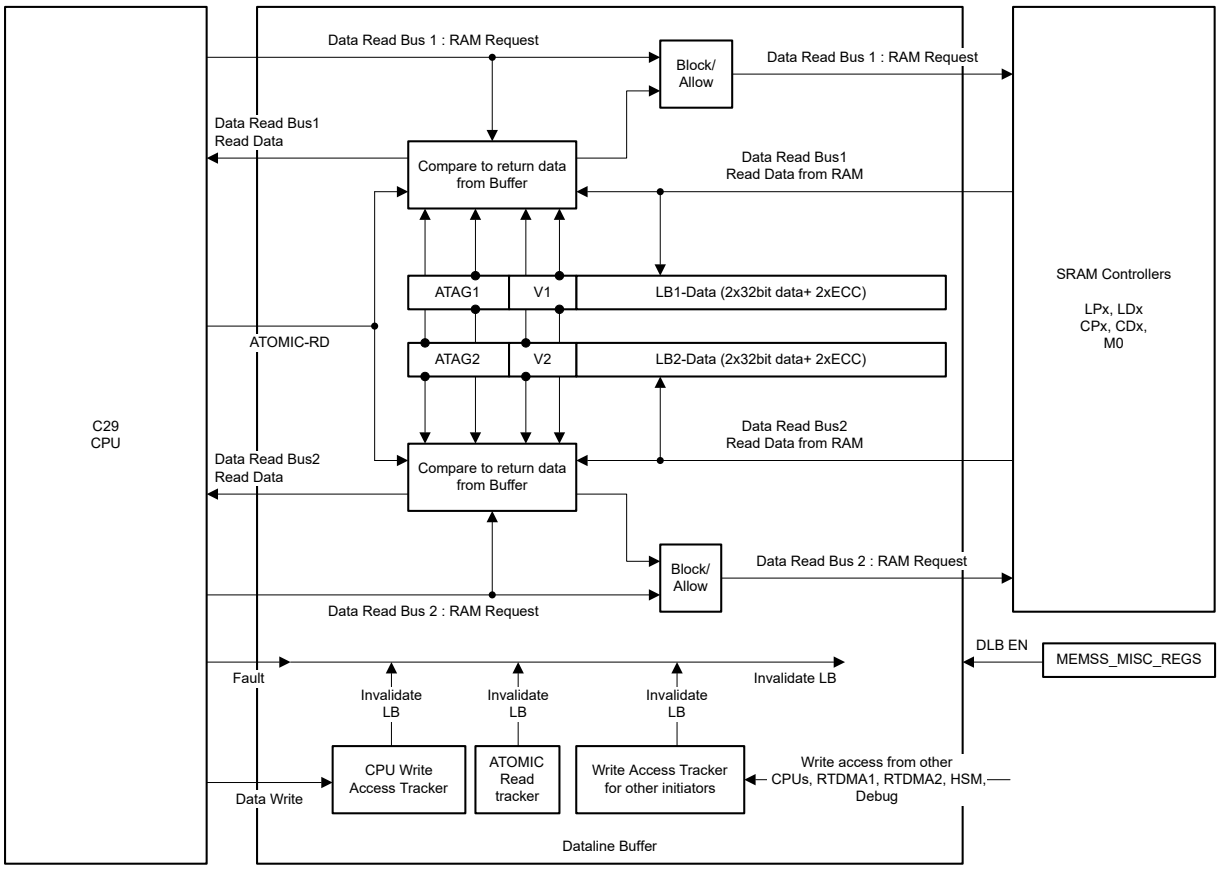


Figure 3-20. Dataline Buffer

LB1 is loaded with data returned on data read bus 1, while LB2 is loaded with data from data read bus 2. The width of line buffer is 64 bits and has an associated valid bit (Vx) and address tag (ATAGx). ATAGx is a 64-bit aligned address of the data being held in the dataline buffer. The memory controller returns 64 bits of data regardless of the read access size. On every read access, the read address is compared with the ATAGx of LB1 and LB2. When the tag matches, data is returned from the corresponding dataline buffer. Read access to the memory controller is blocked at that time.

Initiators other than the CPU have access to RAM on the device. The dataline buffer is updated on CPU reads only, and the data in the buffer does not hold the latest data if another initiator updated the same address already. Write accesses from other initiators are tracked local to the dataline buffer, as shown by the "Write Access Tracker from Other Initiators" in Figure 3-20. The dataline buffer is marked as invalid upon write access completion from another initiator if the address tag matched the write address. The CPU reads the old data until the time at which the buffer is marked as invalid using the Vx bit.

The CPU can issue a read and write to the same address simultaneously, in which case the dataline buffer detects this and invalidates the dataline buffer when there is an address tag match. An address tag match to a write address does not update the dataline buffer locally. For this, the write is performed to memory first then a read is initiated to fetch the latest data. This is shown by "CPU Write Access Tracker" in Figure 3-20.

The ECC check is done within the CPU. When there is an uncorrectable ECC error detected, the CPU enters a fault state. Dataline buffers of a CPU are invalidated upon any CPU fault.

Dataline buffer needs to be disabled using MEM_DLB_CONFIG register before memory initialization or running test mode to invalidate last buffered data.

3.10.4.9 HSM Sync Bridge

HSM Subsystem has two initiators, M4 core and HSM-RTDMA. HSM Sync bridge connects the C29 peripherals and memories to HSM subsystem.

Figure 3-21 consists of MPU (Memory Protection Unit), Data line buffer and Safe interconnect.

MPU configuration can only be done by CPU1 Link0/1/2 using the SYNCBRIDGEMPU Registers.

Read access going into C29 system memories must be 64 bit reads with reads buffered in dataline buffer improving throughput as compared to two 32-bit reads. Although read access to C29 system peripherals are 32-bit reads and is not buffered into dataline buffer.

Write access from other initiators are tracked and dataline buffer is invalidated if the address of the write matches address tag of line buffer similar to C29CPU dataline buffer.

ECC Checks are also performed in the sync bridge as part of the safe interconnect and errors are sent to HSM-ESM (Error Signaling Module).

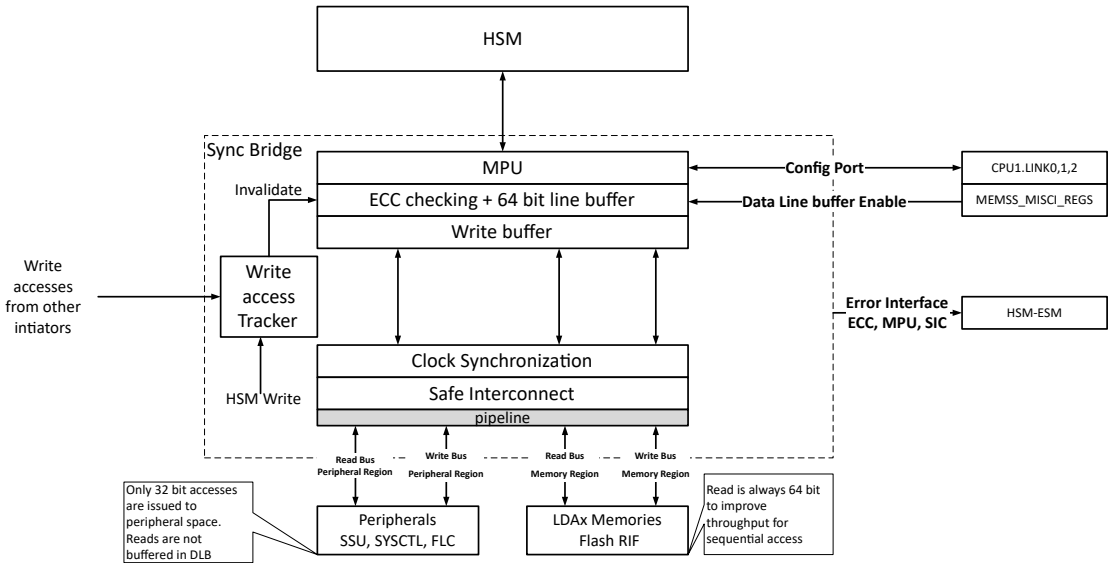


Figure 3-21. HSM Sync Bridge Block Diagram

3.10.4.10 Access Bridges

3.10.4.10.1 Debug Access Bridge

The debug access bridge arbitrates and connects one set of read and write ports from a CPU to the MEMSS memory controller using static select arbitration as described in Section 3.10.6. This is common to the entire system. Arbitration and security access filtering can take up to two cycles, and debug accesses through this bridge are a two wait state minimum.

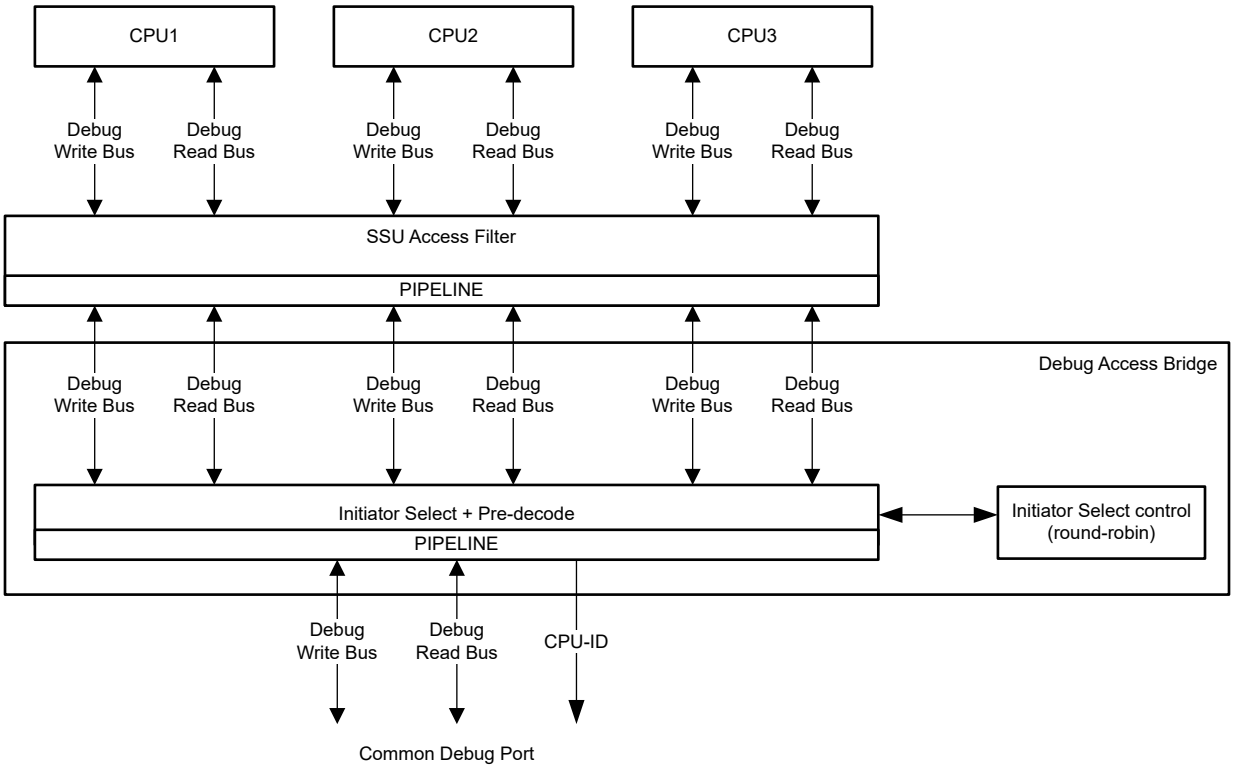


Figure 3-22. Debug Access Bridge

The initiator select connects one of the CPU buses to a common debug port, so the output buses cannot have accesses from more than one initiator at a given time. Debug read and write accesses identified as illegal are blocked in the debug access bridge.

Common debug access ports have a CPU-ID to indicate the origin of the current access. This ID can be used to implement access filtering, if needed. Following are the CPU-ID assignments:

- CPU1: 0001
- CPU2: 0010
- CPU3: 0011

The two types of memory regions have access filter handling :

1. Peripheral and uniquely memory-mapped memories
   a. The SSU handles access filtering.
2. Tightly-coupled CPU peripherals (that is, PIPE, ERAD, and the CPU). These are mapped at some address respective to the CPU memory-map, and only the respective CPU has access to that memory (access checked using CPU-ID).
   a. The SSU access filter always allows access to the ERAD region.

3.10.4.10.2 Global Access Bridge

The global access bridge arbitrates accesses in a round-robin priority. Data access to RAM is accessible from all CPUs, with 9 CPU ports on the RAM controller (two data read ports and one data write port for each CPU). Data accesses are optimized for two CPUs, connected to the MEMSS memory controller directly. Accesses from the remaining CPUs are arbitrated and merged into a generic CPU port with two data read and one data write port. These ports are connected to only one CPU at any given point in time. Figure 3-23 depicts how the LDAn memories are optimized for zero wait states by connecting CPU1 and CPU2 to the direct access port and CPU3 to the global bridge.

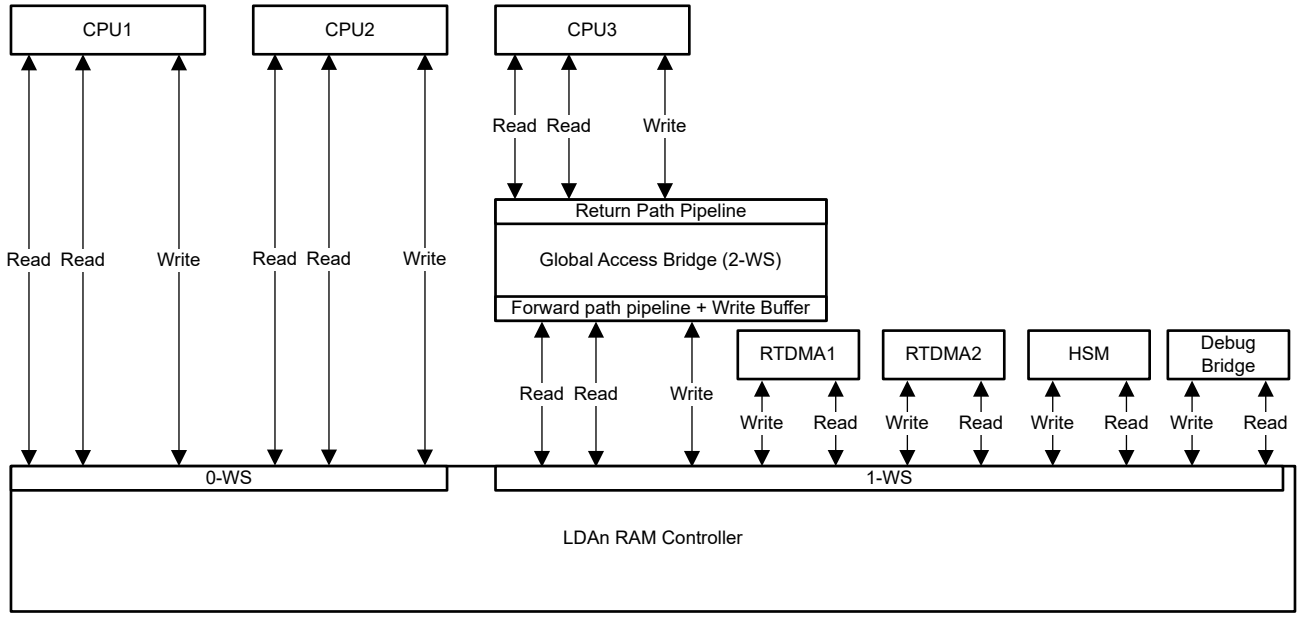


Figure 3-23. Global Access Bridge for LDAn RAMs

At any given time, accesses from one CPU are routed to the other side of the bridge. This global access bridge has only one set of output ports; hence, two different CPUs from bridge cannot access two memory instances concurrently. When a specific CPU is granted access, all active accesses from that CPU are placed into memory simultaneously, helping to support the throughput as shown in Figure 3-24.

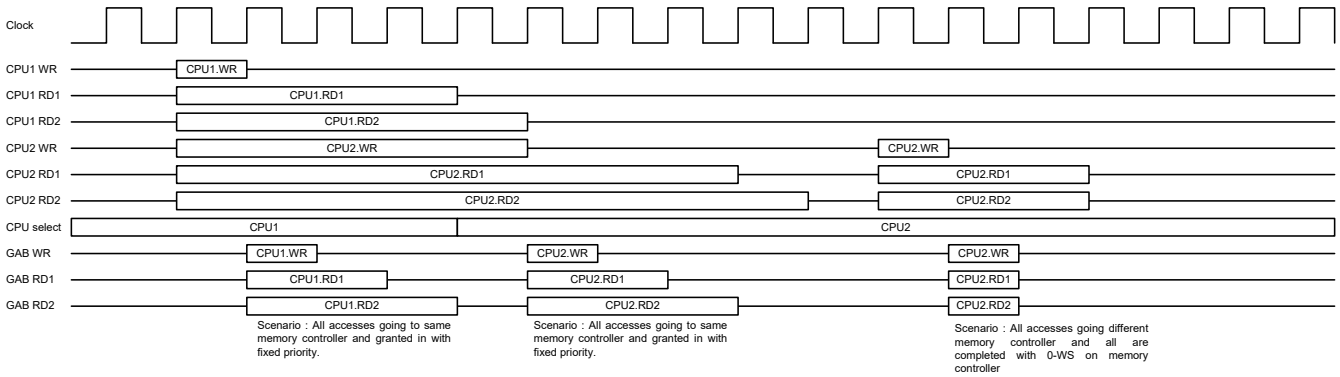


Figure 3-24. Global Access Bridge Arbitration

The global access bridges available to each RAM are shown by Figure 3-13. The global access bridges support atomic operations and posted write with the write buffer. The posted write support is common for all CPUs.

3.10.4.10.3 Program Access Bridge

Figure 3-25 shows the program access bridge, which bridges the 128-bit bus of the CPU and the 64-bit bus of the LDx and CDx memories. The program access bridge is only active when the program access is issued to LDx and CDx memories. The bridge issues two 64-bit accesses in place of one 128-bit access from the CPU. Data is then locally buffered to accumulate 128-bits before returning to the CPU.

Program accesses to LPx and CPx memories bypass the program bridge and are transparent. If the address falls on the second half of the 128-bit word only one access is issued, with the lower 64 bits of the data word ignored by the CPU.

Note

Prefetch access is not supported for M0 RAM from any CPU.

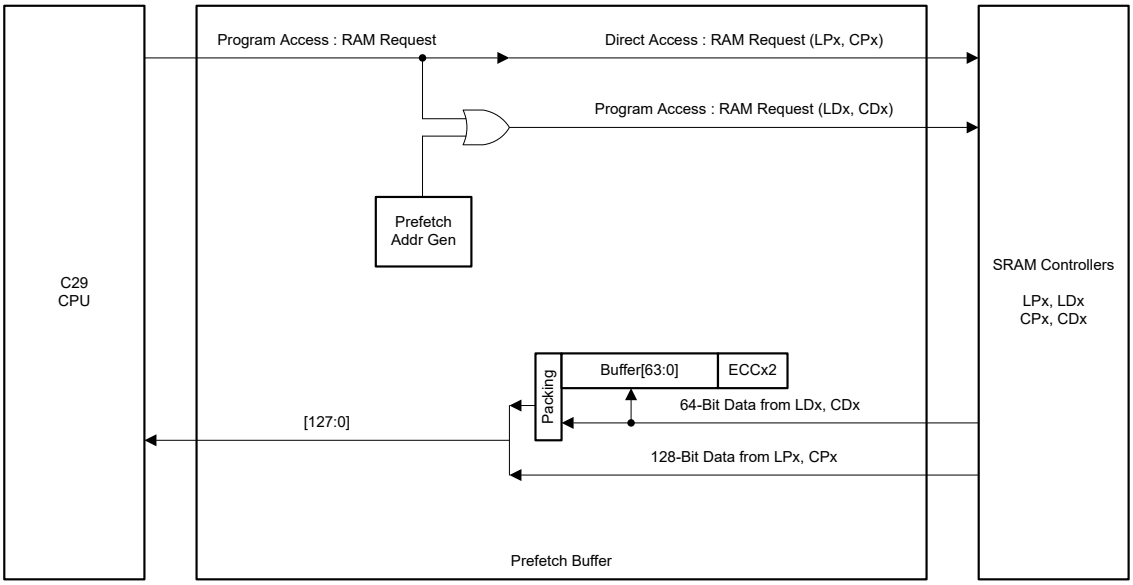


Figure 3-25. Program Access Bridge

3.10.5 ROM

This section details the architecture and configuration of the MEMSS for ROM. Each CPU has dedicated ROM. Program and data accesses are supported and are one wait state. Multiple ROM banks are interleaved to form 256-bit words. ECC is implemented with a 64-bit granularity, which is the same as Flash. A prefetch buffer compensates for the one wait state latency. There is a local 256-bit dataline buffer that holds the last data read from ROM.

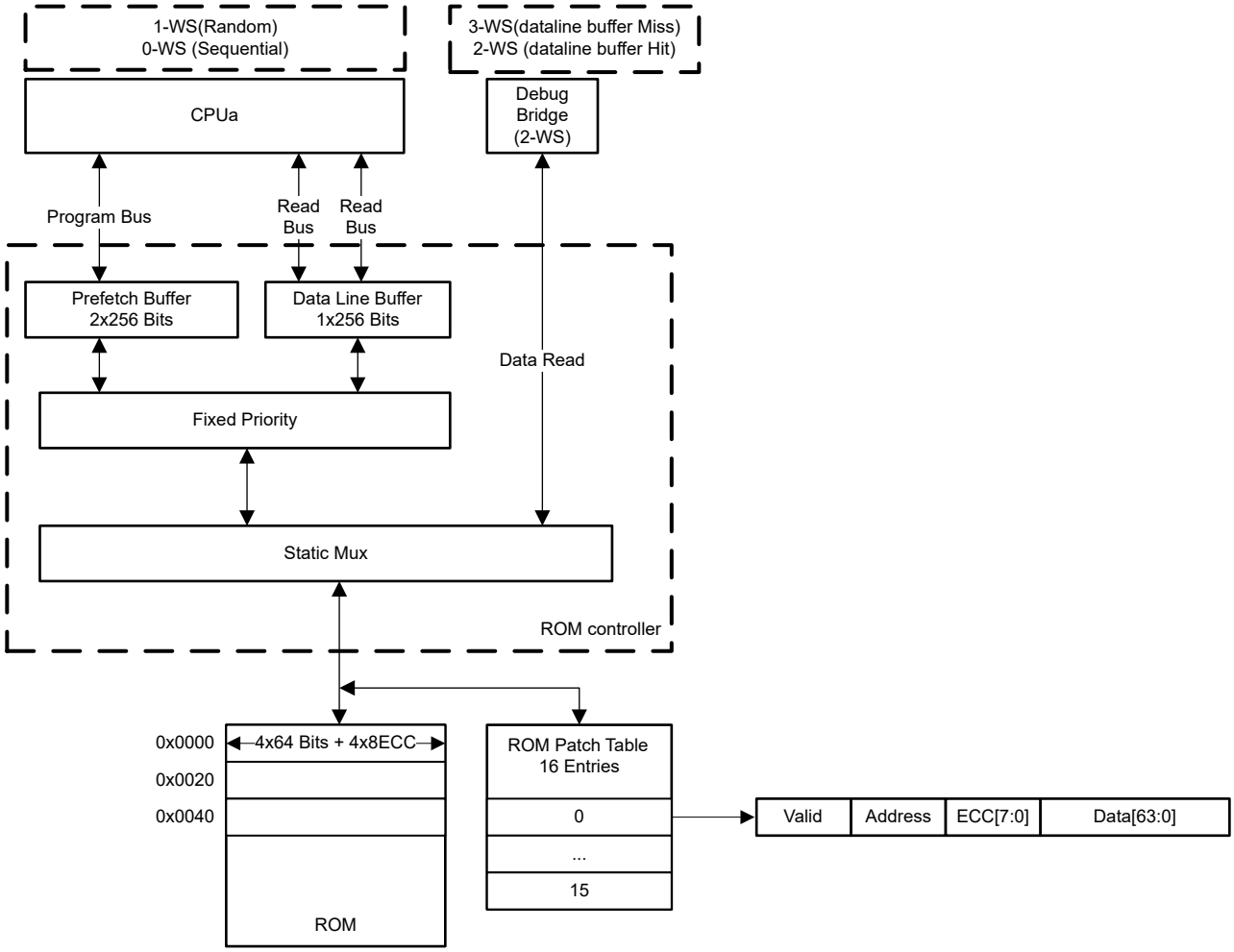


Figure 3-26. ROM Controller

The following is the round-robin priority order:

1. CPUa Access
2. Debug Access

3.10.5.1 ROM Dataline Buffer

The ROM memory controller has a 256-bit dataline buffer. The latest CPU data access from ROM is stored in the dataline buffer. ECC bits are stored with data. Program accesses from the CPU do not affect the dataline buffer. During a simultaneous data read access from bus 1 and bus 2, if the address of the accesses is within the 256-bit aligned address then only one access is issued to ROM. The second access is served from the same data. Debug read access is not buffered in the dataline buffer.

The dataline buffer is flushed when the CPU enters a fault state. The test mode does not affect the dataline buffer operation since ROM is read-only. Depending on the test mode setting, either data or ECC bits are returned.

3.10.5.2 ROM Prefetch

Upon the first program access, data is fetched from memory and presented to the CPU. On every access to ROM, 256 bits are fetched from the address that is aligned to the 256-bit boundary. The prefetch buffer can store up to two 256-bit words and stores both ECC and data. The ROM memory controller prefetches data from the next sequential 256-bit boundary-aligned address as long as there is an empty slot in the prefetch buffer. This is known as a prefetch access. Prefetch and program accesses are lower priority than data accesses as shown in Arbitration section.

This process does not start until ROM receives the first program access. After reset, the prefetch logic is idle until a program access with a discontinuity is received. This discontinuity is indicated by the CPU.

The prefetch buffer is flushed whenever a CPU program access with a discontinuity anywhere on the memory map occurs, not just when ROM is accessed.

3.10.6 Arbitration

The MEMSS uses a three-level arbitration scheme. The first level of arbitration is the fixed priority arbiter, which arbitrates accesses originating from within the same initiator. The following is the priority for the CPU accesses:

1. Write
2. Read Bus 1
3. Read Bus 2
4. Program Access (applies to ROM)
5. Prefetch Access (generated locally and applies to ROM)

Initiators other than the CPU have only one read and one write bus, arbitrated with the following priority:

1. Write
2. Read Bus

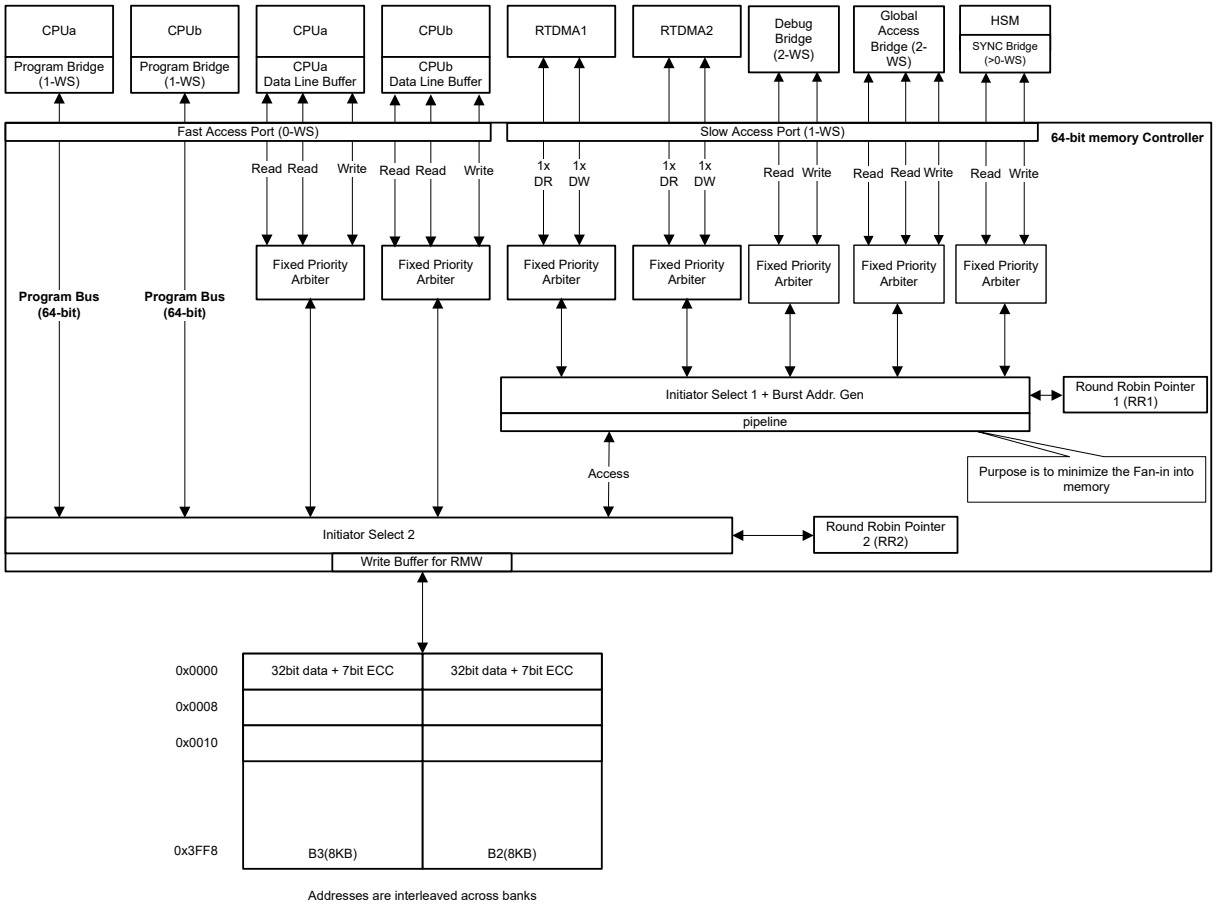


Figure 3-27. Three-Level Arbitration Scheme

The second and third level of arbitration controls the initiator selects. Initiator select 1 and 2 are controlled by independent round-robin pointers RR1 and RR2 respectively. RR1 arbitrates accesses connected to slow access ports. When access is granted by RR1, the access is pipelined and then passes to initiator select 2. Next, RR2 arbitrates accesses from initiator select 1 and fast access ports.

The following tables help to illustrate the round-robin arbitration using 64-bit memory controller as an example and same applies to 128-bit memory controller.

Table 3-9. Example 1

Cycle	CPUa	RTDMA1	RTDMA2	Round-Robin Pointer 2	Round-Robin Pointer 1	Access Granted at Initiator Select 1	Memory Access
1	CPUa Read Access 1	RTDMA1 Read Access 1	RTDMA2 Read Access 1	CPUa	RTDMA1		CPUa Read Access 1
2	CPUa Read Access 2	RTDMA1 Read Access 1	RTDMA2 Read Access 1	CPUa	RTDMA1	RTDMA1 Read Access 1	CPUa Read Access 2
3		RTDMA1 Read Access 1	RTDMA2 Read Access 1	Initiator Select 1	RTDMA1	RTDMA1 Read Access 1	RTDMA1 Read Access1
4			RTDMA2 Read Access 1	Initiator Select 1	RTDMA2		
5			RTDMA2 Read Access 1	Initiator Select 1	RTDMA2	RTDMA2 Read Access 1	RTDMA2 Read Access1

Table 3-9. Example 1 (continued)

Cycle	CPUa	RTDMA1	RTDMA2	Round-Robin Pointer 2	Round-Robin Pointer 1	Access Granted at Initiator Select 1	Memory Access
6				Initiator Select 1	RTDMA2		
7	CPUa Read Access 3			Initiator Select 1	RTDMA2		
8	CPUa Read Access 3			CPUa	RTDMA2		CPUa Read Access 3
9				CPUa	RTDMA2		

Table 3-10. Example 2

Cycle	CPUa	RTDMA1	RTDMA2	Round-Robin Pointer 2	Round-Robin Pointer 1	Access Granted at Initiator Select 1	Memory Access
1	CPUa Read Access 1	RTDMA1 Read Access 1	RTDMA2 Read Access 1	CPUa	RTDMA1		CPUa Read Access 1
2	CPUa Read Access 2	RTDMA1 Read Access 1	RTDMA2 Read Access 1	CPUa	RTDMA1	RTDMA1 Read Access 1	CPUa Read Access 2
3	CPUa Read Access 3	RTDMA1 Read Access 1	RTDMA2 Read Access 1	Initiator Select 1	RTDMA1	RTDMA1 Read Access 1	RTDMA1 Read Access 1
4	CPUa Read Access 3		RTDMA2 Read Access 1	CPUa	RTDMA2		CPUa Read Access 3
5	CPUa Read Access 4		RTDMA2 Read Access 1	CPUa	RTDMA2	RTDMA2 Read Access 1	CPUa Read Access 4
6			RTDMA2 Read Access 1	Initiator Select 1	RTDMA2		RTDMA2 Read Access 1
7				Initiator Select 1	RTDMA2		
8				Initiator Select 1	RTDMA2		
9				Initiator Select 1	RTDMA2		

The default initiator selects at reset is set such that the primary CPU has access to the memory (that is, the primary CPU has access to LPx, LDx, CDx, CPx, and M0 by default).

The round-robin pointer is changed when triggered by a pending request from an initiator other than the one currently granted access. The pointer is changed to the next initiator with priority, which takes 1 cycle. This occurs for each round-robin pointer, with greater weight given to initiators coming from fast access ports.

This two-level arbitration scheme makes sure that no initiator is stalled forever except for atomic and burst operations. Back-to-back accesses or multiple simultaneous accesses from an initiator does not block the other initiator from getting granted access. Essentially, when the round-robin pointer is set to an initiator, a pending request from any other initiator forces the round-robin pointer to switch, so there is fair arbitration without blocking an initiator.

For example, if CPUa and CPUb are initiating back-to-back read accesses the round-robin pointer switches between CPUa and CPUb every cycle so accesses from CPUa and CPUb are served alternatively one after the other.

Another example is if CPUa initiates four possible accesses (program read, data read 1, data read 2, and data write) and the RTDMA is initiating back-to-back read accesses at the same time. If the current pointer is set to CPUa, only the CPUa data write access gets granted before the pointer is switched to RTDMA to serve the read request. The pointer is then switched back to service CPUa program read, and at this time both the new RTDMA request as well as the CPUa data read 1 and 2 are pending. The pointer is switched back to CPUa to serve the data read 1 access and then CPUa data read 2 access.

3.10.7 Test Modes

Test modes can be used for implementing diagnostics tests on memory controllers. These modes enable error injection in data and ECC. The memory controller supports three modes of operation:

- Mode 0: Normal mode of operation
- Mode 1: Write to memory does not update the ECC, only data bits are updated (reading the ECC check is disabled)
- Mode 2: Reading an address returns ECC bits of the data at that address. Writing updates the ECC bits, while the data bits remain unchanged (reading the ECC check is disabled)

In mode 2, only 32-bit reads are supported. The ECC bits appear in byte 0 position of the read data.

The following is an example sequence to inject faults into RAM to implement diagnostics tests:

1. Write data to an address in mode 0 (both data and ECC get written to memory)
2. To induce errors on the data, set the mode to 1; to induce errors on ECC, set the mode to 2
3. Set the mode back to 0
4. Read the same address, with handling available if there is an error

ECC checks are performed in the CPU. CPU has built-in self-test controller and fault emulation to implement diagnostics. The MEMSS test mode is to implement diagnostic tests on ECC logic used for read-modify-write operations local to the memory controller.

Note

Test mode is not a global setting. Each memory controller has a separate TESTMODE control.

The test mode can be enabled by CPU1 code running from LINK0, LINK1, and LINK2. RAM test mode takes effect only for CPU1 accesses. Accesses from other initiators are serviced in the normal mode of operation, irrespective of the RAM test mode setting. Implementing ECC diagnostics for read-modify-write from CPU1 is sufficient since ECC checking logic is common for all initiators. A software handshake can be implemented in case RAM diagnostics are run on shared memory addresses.

Note

The dataline buffer is recommended to be disabled when using test modes to avoid RAM test data from getting buffered.

Real-time interrupts can still be enabled during diagnostics. In this scenario, an atomic sequence can be used to make sure the real-time interrupt is taken only after the dataline buffer is enabled and RAM is configured to mode 0.

3.10.8 Emulation Mode

Safe interconnect checks are disabled for debug accesses. Debug accesses that are less than 32 bits cause a read-modify-write operation. Any errors during this operation do not trigger an NMI, so as to avoid an NMIWD timeout while the CPU is halted. A status is set in the debug controller for CCS to read. An arbitration lock to a CPU is released if that CPU is halted. The RTDMA completes the burst if enabled before being halted.

Debug accesses are tracked as part of the dataline buffer and the buffer is flushed during debug writes.

3.11 System Control Register Configuration Restrictions

Memory-mapped registers in System Control operate on INTOSC1 clock domain, hence any CPU writes to these registers requires delay in between subsequent writes otherwise second write can be lost. The application needs to take this into consideration and check the **SYNCBUSY** and **SYNCBUSYWD** status registers after every write to the registers that are mentioned in Table 3-11.

Table 3-11. System Control Registers Impacted

Registers Requiring Check After Every Write
SYSCLKDIVSEL
SYSPLLCTL1
SYSPLLMULT
PERCLKDIVSEL
ETHERCATCLKCTL
CLBCLKCTL
MCANCLKDIVSEL
WDCR
XCLKOUTDIVSEL
XTALCR
CLKSRCCTL1
CLKSRCCTL2
CLKSRCCTL3
CPU1/2/3.TMR2CLKCTL

3.12 Software

3.12.1 SYSCTL Registers to Driverlib Functions

Table 3-12. SYSCTL Registers to Driverlib Functions

File	Driverlib Function
DEVCFGLOCK1	
sysctl.h	SysCtl_lockAllPeriphConfigRegisters
DEVCFGLOCK2	
-	
DEVCFGLOCK3	
sysctl.h	SysCtl_lockAllPeriphConfigRegisters
DEVCFGLOCK4	
sysctl.h	SysCtl_lockAllPeriphConfigRegisters
DEVCFGLOCK5	
sysctl.h	SysCtl_lockAllPeriphConfigRegisters
PARTIDL	
sysctl.c	SysCtl_getDeviceParametric
PARTIDH	
sysctl.c	SysCtl_getDeviceParametric
REVID	
sysctl.h	SysCtl_getDeviceRevision
MCUCNF1	
sysctl.c	SysCtl_emulateDevice
MCUCNF2	
sysctl.c	SysCtl_emulateDevice
MCUCNF4	
sysctl.c	SysCtl_emulateDevice
MCUCNF7	
sysctl.c	SysCtl_emulateDevice
MCUCNF10	
sysctl.c	SysCtl_emulateDevice
MCUCNF13	
sysctl.c	SysCtl_emulateDevice
MCUCNF14	
sysctl.c	SysCtl_emulateDevice
MCUCNF16	
sysctl.c	SysCtl_emulateDevice
MCUCNF17	
sysctl.c	SysCtl_emulateDevice
MCUCNF18	
sysctl.c	SysCtl_emulateDevice
MCUCNF19	
sysctl.c	SysCtl_emulateDevice
MCUCNF23	
sysctl.c	SysCtl_emulateDevice
MCUCNF26	
sysctl.c	SysCtl_emulateDevice

Table 3-12. SYSCTL Registers to Driverlib Functions (continued)

File	Driverlib Function
MCUCNF31	
sysctl.c	SysCtl_emulateDevice
MCUCNF64	
sysctl.c	SysCtl_emulateDevice
MCUCNF65	
sysctl.c	SysCtl_emulateDevice
MCUCNF74	
sysctl.c	SysCtl_emulateDevice
MCUCNF76	
sysctl.c	SysCtl_emulateDevice
MCUCNF78	
sysctl.c	SysCtl_emulateDevice
MCUCNF79	
sysctl.c	SysCtl_emulateDevice
MCUCNF81	
sysctl.c	SysCtl_emulateDevice
MCUCNFLOCK1	
sysctl.h	SysCtl_lockAllMCUCNFRegisters
MCUCNFLOCK3	
sysctl.h	SysCtl_lockAllMCUCNFRegisters
LSEN	
sysctl.h	SysCtl_disableLockStep
EPWMXLINKCFG	
sysctl.h	SysCtl_enableEPWMXLINK
sysctl.h	SysCtl_disableEPWMXLINK
SICCONFIG	
sysctl.h	SysCtl_enableSafetyInterconnect
sysctl.h	SysCtl_disableSafetyInterconnect
RSTSTAT	
sysctl.h	SysCtl_isCPU2Reset
sysctl.h	SysCtl_isCPU3Reset
LPMSTAT	
sysctl.h	SysCtl_getCPU2LPMStatus
sysctl.h	SysCtl_getCPU3LPMStatus
TAP_STATUS	
-	
TAP_CONTROL	
-	
DEVLIFECYCLE	
-	
SDFMTYPE	
-	
SYNCSELECT	
sysctl.h	SysCtl_setSyncOutputConfig
ADCSOCOUTSELECT	

Table 3-12. SYSCTL Registers to Driverlib Functions (continued)

File	Driverlib Function
sysctl.h	SysCtl_enableExtADCSOCSource
sysctl.h	SysCtl_disableExtADCSOCSource
ADCSOCOUTSELECT1	
sysctl.h	SysCtl_enableExtADCSOCSource
sysctl.h	SysCtl_disableExtADCSOCSource
SYNCSOCLOCK	
-	
HSMTOCPU_STS1	
-	
HSMTOCPU_STS2	
-	
HSM_SECURE_BOOT_INFO_REG0	
-	
HSM_SECURE_BOOT_INFO_REG1	
-	
HSM_SECURE_BOOT_INFO_REG2	
-	
HSM_SECURE_BOOT_INFO_REG3	
-	
HSM_SECURE_BOOT_INFO_REG4	
-	
HSM_SECURE_BOOT_INFO_REG5	
-	
HSM_SECURE_BOOT_INFO_REG6	
-	
HSM_SECURE_BOOT_INFO_REG7	
-	
SOC_SECURE_BOOT_INFO_REG0	
-	
SOC_SECURE_BOOT_INFO_REG1	
-	
SOC_SECURE_BOOT_INFO_REG2	
-	
SOC_SECURE_BOOT_INFO_REG3	
-	
SOC_SECURE_BOOT_INFO_REG4	
-	
SOC_SECURE_BOOT_INFO_REG5	
-	
SOC_SECURE_BOOT_INFO_REG6	
-	
SOC_SECURE_BOOT_INFO_REG7	
-	
CLKCFGLOCK1	
sysctl.h	SysCtl_lockAllClockConfigRegisters

Table 3-12. SYSCTL Registers to Driverlib Functions (continued)

File	Driverlib Function
CLKSRCCTL1	
sysctl.c	SysCtl_selectXTAL
sysctl.c	SysCtl_selectXTALSingleEnded
sysctl.c	SysCtl_selectOscSource
sysctl.c	SysCtl_getClock
sysctl.c	SysCtl_setClock
CLKSRCCTL2	
clockconfig.h	__attribute__
CLKSRCCTL3	
clockconfig.h	__attribute__
SYSPLLCTL1	
sysctl.c	SysCtl_getClock
sysctl.c	SysCtl_setClock
SYSPLLMULT	
sysctl.c	SysCtl_getClock
sysctl.c	SysCtl_setClock
SYSPLLSTS	
sysctl.c	SysCtl_setClock
SYSCLKDIVSEL	
sysctl.c	SysCtl_getClock
sysctl.c	SysCtl_setClock
PERCLKDIVSEL	
clockconfig.h	__attribute__
clockconfig.h	__attribute__
clockconfig.h	__attribute__
clockconfig.h	__attribute__
XCLKOUTDIVSEL	
clockconfig.h	__attribute__
HSMCLKDIVSEL	
clockconfig.h	__attribute__
MCANCLKDIVSEL	
clockconfig.h	__attribute__
CLBCLKCTL	
clockconfig.h	__attribute__
MCDCR	
clockconfig.h	__attribute__
clockconfig.h	__attribute__
clockconfig.h	__attribute__
clockconfig.h	__attribute__
clockconfig.h	__attribute__
clockconfig.h	__attribute__
X1CNT	
clockconfig.h	__attribute__
clockconfig.h	__attribute__
sysctl.c	SysCtl_pollX1Counter

Table 3-12. SYSCTL Registers to Driverlib Functions (continued)

File	Driverlib Function
XTALCR	
clockconfig.h	__attribute__
clockconfig.h	__attribute__
sysctl.c	SysCtl_selectXTAL
sysctl.c	SysCtl_selectXTALSingleEnded
sysctl.c	SysCtl_setClock
XTALCR2	
sysctl.c	SysCtl_selectXTAL
ETHERCATCLKCTL	
clockconfig.h	__attribute__
ETHERCATCTL	
sysctl.h	SysCtl_enableEthercatI2Cloopback
sysctl.h	SysCtl_disableEthercatI2Cloopback
SYNCBUSY	
sysctl.c	SysCtl_pollSyncBusy
ESMXRSNCTL	
sysctl.h	SysCtl_enableESMResetCauses
sysctl.h	SysCtl_disableESMResetCauses
EPWM1	
-	
EPWM2	
-	
EPWM3	
-	
EPWM4	
-	
EPWM5	
-	
EPWM6	
-	
EPWM7	
-	
EPWM8	
-	
EPWM9	
-	
EPWM10	
-	
EPWM11	
-	
EPWM12	
-	
EPWM13	
-	
EPWM14	

Table 3-12. SYSCTL Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
EPWM15	
-	
EPWM16	
-	
EPWM17	
-	
EPWM18	
-	
HRCAL0	
-	
HRCAL1	
-	
HRCAL2	
-	
ECAP1	
-	
ECAP2	
-	
ECAP3	
-	
ECAP4	
-	
ECAP5	
-	
ECAP6	
-	
EQEP1	
-	
EQEP2	
-	
EQEP3	
-	
EQEP4	
-	
EQEP5	
-	
EQEP6	
-	
SDFM1	
-	
SDFM2	
-	
SDFM3	
-	

Table 3-12. SYSCTL Registers to Driverlib Functions (continued)

File	Driverlib Function
SDFM4	
-	
UARTA	
-	
UARTB	
-	
UARTC	
-	
UARTD	
-	
UARTE	
-	
UARTF	
-	
SPIA	
-	
SPIB	
-	
SPIC	
-	
SPID	
-	
SPIE	
-	
I2CA	
-	
I2CB	
-	
PMBUSA	
-	
LINA	
-	
LINB	
-	
MCANA	
-	
MCANB	
-	
MCANC	
clockconfig.h	__attribute__
MCAND	
-	
MCANE	
-	
MCANF	

Table 3-12. SYSCTL Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
ADCA	
-	
ADCB	
-	
ADCC	
-	
ADCD	
-	
ADCE	
-	
CMPSS1	
-	
CMPSS2	
-	
CMPSS3	
-	
CMPSS4	
-	
CMPSS5	
-	
CMPSS6	
-	
CMPSS7	
-	
CMPSS8	
-	
CMPSS9	
-	
CMPSS10	
-	
CMPSS11	
-	
CMPSS12	
-	
DACA	
-	
DACB	
-	
CLB1	
-	
CLB2	
-	
CLB3	
-	

Table 3-12. SYSCTL Registers to Driverlib Functions (continued)

File	Driverlib Function
CLB4	
-	
CLB5	
-	
CLB6	
-	
FSITXA	
-	
FSITXB	
-	
FSITXC	
-	
FSITXD	
-	
FSIRXA	
-	
FSIRXB	
-	
FSIRXC	
-	
FSIRXD	
-	
DCC1	
-	
DCC2	
-	
DCC3	
-	
ETHERCATA	
-	
EPG1	
-	
SENT1	
-	
SENT2	
-	
SENT3	
-	
SENT4	
-	
SENT5	
-	
SENT6	
-	
ADCCHECKER1	

Table 3-12. SYSCTL Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
ADCCHECKER2	
-	
ADCCHECKER3	
-	
ADCCHECKER4	
-	
ADCCHECKER5	
-	
ADCCHECKER6	
-	
ADCCHECKER7	
-	
ADCCHECKER8	
-	
ADCCHECKER9	
-	
ADCCHECKER10	
-	
ADCSEAGGRCPU1	
-	
ADCSEAGGRCPU2	
-	
ADCSEAGGRCPU3	
-	
RTDMA1CH	
-	
RTDMA2CH	
-	
WADI1	
-	
WADI2	
-	
INPUTXBARFLAGS	
-	
OUTPUTXBARFLAGS	
-	
DLTFIFOREGS	
-	
ADC_GLOBAL_REGS	
-	
ERROR_AGGREGATOR	
-	
ESM	
sysctl.h	SysCtl_enableESMResetCauses

Table 3-12. SYSCTL Registers to Driverlib Functions (continued)

File	Driverlib Function
sysctl.h	SysCtl_disableESMResetCauses
PARITY_TEST	
-	
CPUPERCFGLOCK1	
sysctl.h	SysCtl_lockAllPeriphClockRegisters
CPUPERCFGLOCK2	
-	
PCLKCR0	
sysctl.h	SysCtl_enablePeripheral
sysctl.h	SysCtl_disablePeripheral
PCLKCR1	
-	See PCLKCR0
PCLKCR2	
-	See PCLKCR0
PCLKCR3	
-	See PCLKCR0
PCLKCR4	
-	See PCLKCR0
PCLKCR6	
-	See PCLKCR0
PCLKCR7	
-	See PCLKCR0
PCLKCR8	
-	See PCLKCR0
PCLKCR9	
-	See PCLKCR0
PCLKCR10	
-	See PCLKCR0
PCLKCR13	
-	See PCLKCR0
PCLKCR14	
-	See PCLKCR0
PCLKCR16	
-	See PCLKCR0
PCLKCR17	
-	
PCLKCR18	
-	
PCLKCR19	
-	
PCLKCR20	
-	
PCLKCR21	
-	
PCLKCR23	

Table 3-12. SYSCTL Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
PCLKCR25	
-	
PCLKCR27	
-	
PCLKCR28	
-	
PCLKCR30	
-	
PCLKCR32	
-	
SOFTPRES0	
sysctl.h	SysCtl_resetPeripheral
SOFTPRES1	
-	See SOFTPRES0
SOFTPRES2	
-	See SOFTPRES0
SOFTPRES3	
-	See SOFTPRES0
SOFTPRES4	
-	See SOFTPRES0
SOFTPRES6	
-	See SOFTPRES0
SOFTPRES7	
-	See SOFTPRES0
SOFTPRES8	
-	See SOFTPRES0
SOFTPRES9	
-	See SOFTPRES0
SOFTPRES10	
-	See SOFTPRES0
SOFTPRES13	
-	See SOFTPRES0
SOFTPRES14	
-	See SOFTPRES0
SOFTPRES16	
-	See SOFTPRES0
SOFTPRES17	
-	
SOFTPRES18	
-	
SOFTPRES19	
-	
SOFTPRES20	
-	

Table 3-12. SYSCTL Registers to Driverlib Functions (continued)

File	Driverlib Function
SOFTPRES21	
-	
SOFTPRES23	
-	
SOFTPRES25	
-	
SOFTPRES27	
-	
SOFTPRES28	
-	
SOFTPRES30	
-	
SOFTPRES32	
-	
PARITY_TEST_ALT1	
-	
CPUSYSLOCK1	
-	
CPUID	
sysctl.h	SysCtl_getCPUID
LPMCR	
sysctl.h	SysCtl_enterIdleMode
sysctl.h	SysCtl_enterStandbyMode
sysctl.h	SysCtl_setStandbyQualificationPeriod
CMPSSLPMSEL	
sysctl.h	SysCtl_enableLPMWakeupPin
sysctl.h	SysCtl_disableLPMWakeupPin
GPIOLPMSEL0	
-	
GPIOLPMSEL1	
-	
TMR2CLKCTL	
clockconfig.h	__attribute__
RESCCLR	
sysctl.h	SysCtl_clearResetCause
watchdog.h	staticinlinevoidSysCtl_clearWatchdogResetStatus
RESC	
sysctl.h	SysCtl_clearResetCause
watchdog.h	staticinlineboolSysCtl_getWatchdogResetStatus
watchdog.h	staticinlinevoidSysCtl_clearWatchdogResetStatus
MCANWAKESTATUS	
sysctl.h	SysCtl_isMCANWakeStatusSet
sysctl.h	SysCtl_clearMCANWakeStatus
MCANWAKESTATUSCLR	
sysctl.h	SysCtl_clearMCANWakeStatus

Table 3-12. SYSCTL Registers to Driverlib Functions (continued)

File	Driverlib Function
CLKSTOPREQ	
-	
CLKSTOPACK	
-	
USER_REG1_SYSRSN	
sysctl.h	SysCtl_setUserRegister
sysctl.h	SysCtl_getUserRegister
USER_REG2_SYSRSN	
-	
USER_REG1_XRSN	
-	
USER_REG2_XRSN	
-	
USER_REG1_PORESETN	
-	
USER_REG2_PORESETN	
-	
USER_REG3_PORESETN	
-	
USER_REG4_PORESETN	
-	
SIMRESET	
sysctl.h	SysCtl_simulateReset
PARITY_TEST_ALT2	
-	
SCSR	
watchdog.h	staticinlinevoidSysCtl_setWatchdogMode
watchdog.h	staticinlineboolSysCtl_isWatchdogInterruptActive
watchdog.h	staticinlinevoidSysCtl_clearWatchdogOverride
WDCNTR	
watchdog.h	staticinlineuint16_tSysCtl_getWatchdogCounterValue
WDKEY	
watchdog.h	staticinlinevoidSysCtl_serviceWatchdog
watchdog.h	staticinlinevoidSysCtl_enableWatchdogReset
watchdog.h	staticinlinevoidSysCtl_resetWatchdog
SYNCBUSYWD	
-	
WDCR	
watchdog.h	staticinlinevoidSysCtl_disableWatchdog
watchdog.h	staticinlinevoidSysCtl_enableWatchdog
watchdog.h	staticinlinevoidSysCtl_setWatchdogPredivider
watchdog.h	staticinlinevoidSysCtl_setWatchdogPrescaler
WDWCR	
watchdog.h	staticinlinevoidSysCtl_setWatchdogWindowValue

3.12.2 MEMSS Registers to Driverlib Functions

Table 3-13. MEMSS Registers to Driverlib Functions

File	Driverlib Function
LPA0_MEM_CONFIG	
-	
LPA0_MEM_CONFIG_LOCK	
-	
LPA0_MEM_CONFIG_COMMIT	
-	
LPA1_MEM_CONFIG	
-	
LPA1_MEM_CONFIG_LOCK	
-	
LPA1_MEM_CONFIG_COMMIT	
-	
LDA0_MEM_CONFIG	
-	
LDA0_MEM_CONFIG_LOCK	
-	
LDA0_MEM_CONFIG_COMMIT	
-	
LDA1_MEM_CONFIG	
-	
LDA1_MEM_CONFIG_LOCK	
-	
LDA1_MEM_CONFIG_COMMIT	
-	
LDA2_MEM_CONFIG	
-	
LDA2_MEM_CONFIG_LOCK	
-	
LDA2_MEM_CONFIG_COMMIT	
-	
LDA3_MEM_CONFIG	
-	
LDA3_MEM_CONFIG_LOCK	
-	
LDA3_MEM_CONFIG_COMMIT	
-	
LDA4_MEM_CONFIG	
-	
LDA4_MEM_CONFIG_LOCK	
-	
LDA4_MEM_CONFIG_COMMIT	
-	
LDA5_MEM_CONFIG	

Table 3-13. MEMSS Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
LDA5_MEM_CONFIG_LOCK	
-	
LDA5_MEM_CONFIG_COMMIT	
-	
LDA6_MEM_CONFIG	
-	
LDA6_MEM_CONFIG_LOCK	
-	
LDA6_MEM_CONFIG_COMMIT	
-	
LDA7_MEM_CONFIG	
-	
LDA7_MEM_CONFIG_LOCK	
-	
LDA7_MEM_CONFIG_COMMIT	
-	
CPA0_MEM_CONFIG	
-	
CPA0_MEM_CONFIG_LOCK	
-	
CPA0_MEM_CONFIG_COMMIT	
-	
CPA1_MEM_CONFIG	
-	
CPA1_MEM_CONFIG_LOCK	
-	
CPA1_MEM_CONFIG_COMMIT	
-	
CDA0_MEM_CONFIG	
-	
CDA0_MEM_CONFIG_LOCK	
-	
CDA0_MEM_CONFIG_COMMIT	
-	
CDA1_MEM_CONFIG	
-	
CDA1_MEM_CONFIG_LOCK	
-	
CDA1_MEM_CONFIG_COMMIT	
-	
CDA2_MEM_CONFIG	
-	
CDA2_MEM_CONFIG_LOCK	
-	

Table 3-13. MEMSS Registers to Driverlib Functions (continued)

File	Driverlib Function
CDA2_MEM_CONFIG_COMMIT	
-	
CDA3_MEM_CONFIG	
-	
CDA3_MEM_CONFIG_LOCK	
-	
CDA3_MEM_CONFIG_COMMIT	
-	
CDA4_MEM_CONFIG	
-	
CDA4_MEM_CONFIG_LOCK	
-	
CDA4_MEM_CONFIG_COMMIT	
-	
CDA5_MEM_CONFIG	
-	
CDA5_MEM_CONFIG_LOCK	
-	
CDA5_MEM_CONFIG_COMMIT	
-	
CDA6_MEM_CONFIG	
-	
CDA6_MEM_CONFIG_LOCK	
-	
CDA6_MEM_CONFIG_COMMIT	
-	
CDA7_MEM_CONFIG	
-	
CDA7_MEM_CONFIG_LOCK	
-	
CDA7_MEM_CONFIG_COMMIT	
-	
CDA8_MEM_CONFIG	
-	
CDA8_MEM_CONFIG_LOCK	
-	
CDA8_MEM_CONFIG_COMMIT	
-	
CDA9_MEM_CONFIG	
-	
CDA9_MEM_CONFIG_LOCK	
-	
CDA9_MEM_CONFIG_COMMIT	
-	
CDA10_MEM_CONFIG	

Table 3-13. MEMSS Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
CDA10_MEM_CONFIG_LOCK	
-	
CDA10_MEM_CONFIG_COMMIT	
-	
CDA11_MEM_CONFIG	
-	
CDA11_MEM_CONFIG_LOCK	
-	
CDA11_MEM_CONFIG_COMMIT	
-	
M0_MEM_CONFIG	
-	
M0_MEM_CONFIG_LOCK	
-	
M0_MEM_CONFIG_COMMIT	
-	
CPU1_ROM_CONFIG	
-	
CPU1_ROM_CONFIG_LOCK	
-	
CPU1_ROM_CONFIG_COMMIT	
-	
CPU2_ROM_CONFIG	
-	
CPU2_ROM_CONFIG_LOCK	
-	
CPU2_ROM_CONFIG_COMMIT	
-	
CPU3_ROM_CONFIG	
-	
CPU3_ROM_CONFIG_LOCK	
-	
CPU3_ROM_CONFIG_COMMIT	
-	
MEM_DLB_CONFIG	
-	
MEM_DLB_CONFIG_LOCK	
-	
MEM_DLB_CONFIG_COMMIT	
-	
PERI_MEM_TEST_LOCK	
-	
PERI_MEM_TEST_CONTROL	
-	

Table 3-13. MEMSS Registers to Driverlib Functions (continued)

File	Driverlib Function
PARITY_TEST	
-	
CPU1_ROM_PATCH_REG_LOCK	
-	
CPU1_ROM_PATCH_REG_COMMIT	
-	
CPU1_0_ADDR	
-	
CPU1_0_ECC	
-	
CPU1_0_DATA64_L	
-	
CPU1_0_DATA64_H	
-	
CPU1_1_ADDR	
-	
CPU1_1_ECC	
-	
CPU1_1_DATA64_L	
-	
CPU1_1_DATA64_H	
-	
CPU1_2_ADDR	
-	
CPU1_2_ECC	
-	
CPU1_2_DATA64_L	
-	
CPU1_2_DATA64_H	
-	
CPU1_3_ADDR	
-	
CPU1_3_ECC	
-	
CPU1_3_DATA64_L	
-	
CPU1_3_DATA64_H	
-	
CPU1_4_ADDR	
-	
CPU1_4_ECC	
-	
CPU1_4_DATA64_L	
-	
CPU1_4_DATA64_H	

Table 3-13. MEMSS Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
CPU1_5_ADDR	
-	
CPU1_5_ECC	
-	
CPU1_5_DATA64_L	
-	
CPU1_5_DATA64_H	
-	
CPU1_6_ADDR	
-	
CPU1_6_ECC	
-	
CPU1_6_DATA64_L	
-	
CPU1_6_DATA64_H	
-	
CPU1_7_ADDR	
-	
CPU1_7_ECC	
-	
CPU1_7_DATA64_L	
-	
CPU1_7_DATA64_H	
-	
CPU1_8_ADDR	
-	
CPU1_8_ECC	
-	
CPU1_8_DATA64_L	
-	
CPU1_8_DATA64_H	
-	
CPU1_9_ADDR	
-	
CPU1_9_ECC	
-	
CPU1_9_DATA64_L	
-	
CPU1_9_DATA64_H	
-	
CPU1_10_ADDR	
-	
CPU1_10_ECC	
-	

Table 3-13. MEMSS Registers to Driverlib Functions (continued)

File	Driverlib Function
CPU1_10_DATA64_L	
-	
CPU1_10_DATA64_H	
-	
CPU1_11_ADDR	
-	
CPU1_11_ECC	
-	
CPU1_11_DATA64_L	
-	
CPU1_11_DATA64_H	
-	
CPU1_12_ADDR	
-	
CPU1_12_ECC	
-	
CPU1_12_DATA64_L	
-	
CPU1_12_DATA64_H	
-	
CPU1_13_ADDR	
-	
CPU1_13_ECC	
-	
CPU1_13_DATA64_L	
-	
CPU1_13_DATA64_H	
-	
CPU1_14_ADDR	
-	
CPU1_14_ECC	
-	
CPU1_14_DATA64_L	
-	
CPU1_14_DATA64_H	
-	
CPU1_15_ADDR	
-	
CPU1_15_ECC	
-	
CPU1_15_DATA64_L	
-	
CPU1_15_DATA64_H	
-	

3.12.3 CPU Registers to Driverlib Functions

Table 3-14. CPU Registers to Driverlib Functions

File	Driverlib Function
RTINT_STACK_DATA0(i)	
-	
RTINT_STACK_DATA1(i)	
-	
RTINT_STACK_DATA2(i)	
-	
RTINT_STACK_DATA3(i)	
-	
RTINT_STACK_DATA4(i)	
-	
RTINT_STACK_DATA5(i)	
-	
RTINT_STACK_DATA6(i)	
-	
RTINT_STACK_DATA7(i)	
-	
RTINT_STACK_DATA8(i)	
-	
RTINT_STACK_ECC0(i)	
-	
RTINT_STACK_ECC1(i)	
-	
RTINT_STACK_ECC2(i)	
-	
RTINT_STACK_ECC3(i)	
-	
SECCALL_STACK_DATA0(i)	
-	
SECCALL_STACK_DATA1(i)	
-	
SECCALL_STACK_DATA2(i)	
-	
SECSP0	
-	
SECSP1	
-	
SECSP2	
-	
SECSP3	
-	
SECSP4	
-	
SECSP5	

Table 3-14. CPU Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
SECSP6	
-	
SECSP7	
-	
PSP	
-	
WARNPSP	
-	
MAXPSP	
-	
REVISION	
-	
C29_REGS_LOCK	
-	
C29_REGS_COMMIT	
-	
FLTEMU_CONFIG	
-	
FLTEMU_ACCGRPSEL	
-	
FLTEMU_BITSEL	
-	
FLTEMU_ADDR	
-	
TMU_ROM_PAR_FORCE	
-	
SELFTEST_DIAG_DATA0	
-	
SELFTEST_DIAG_DATA1	
-	
SELFTEST_DIAG_DATA2	
-	
SELFTEST_DIAG_ECC	
-	
SELFTEST_DIAG_CONTROL	
-	
SELFTEST_DIAG_STATUS	
-	
SELFTEST_DIAG_STATUS_CLR	
-	

3.12.4 WD Registers to Driverlib Functions

Table 3-15. WD Registers to Driverlib Functions

File	Driverlib Function
SCSR	
-	
CNTR	
-	
KEY	
-	
CR	
-	
WCR	
-	

3.12.5 CPUTIMER Registers to Driverlib Functions

Table 3-16. CPUTIMER Registers to Driverlib Functions

File	Driverlib Function
TIM	
cputimer.h	CPUTimer_getTimerCount
PRD	
cputimer.h	CPUTimer_setPeriod
TCR	
cputimer.c	CPUTimer_setEmulationMode
cputimer.h	CPUTimer_clearOverflowFlag
cputimer.h	CPUTimer_disableInterrupt
cputimer.h	CPUTimer_enableInterrupt
cputimer.h	CPUTimer_reloadTimerCounter
cputimer.h	CPUTimer_stopTimer
cputimer.h	CPUTimer_resumeTimer
cputimer.h	CPUTimer_startTimer
cputimer.h	CPUTimer_getTimerOverflowStatus
TPR	
cputimer.h	CPUTimer_setPreScaler
TPRH	
cputimer.h	CPUTimer_setPreScaler

3.12.6 XINT Registers to Driverlib Functions

Table 3-17. XINT Registers to Driverlib Functions

File	Driverlib Function
1CR	
gpio.c	GPIO_setInterruptPin
gpio.h	GPIO_setInterruptType
gpio.h	GPIO_getInterruptType
gpio.h	GPIO_enableInterrupt
gpio.h	GPIO_disableInterrupt
gpio.h	GPIO_getInterruptCounter

Table 3-17. XINT Registers to Driverlib Functions (continued)

File	Driverlib Function
2CR	
-	See 1CR
3CR	
-	See 1CR
4CR	
-	See 1CR
5CR	
-	See 1CR
1CTR	
gpio.h	GPIO_getInterruptCounter
2CTR	
-	
3CTR	
-	

3.12.7 LPOST Registers to Driverlib Functions

Table 3-18. LPOST Registers to Driverlib Functions

File	Driverlib Function
REVISION	
-	
CONTROL	
-	
CONFIG	
-	
STATUS	
-	
STATUS_CLEAR	
-	
LOCK	
-	
COMMIT	
-	

3.12.8 SYSCTL Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/sysctl

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

3.12.8.1 Missing clock detection (MCD) - SINGLE_CORE

FILE: sysctl_ex1_missing_clock_detection.c

This example demonstrates the missing clock detection functionality and the way to handle it. Once the MCD is simulated by disconnecting the OSCCLK to the MCD module an NMI would be generated. This NMI determines that an MCD was generated due to a clock failure which is handled in the ISR.

Before an MCD the clock frequency would be as per device initialization (200Mhz). Post MCD the frequency would move to 10Mhz or INTOSC1.

The example also shows how we can lock the PLL after missing clock, detection, by first explicitly switching the clock source to INTOSC1, resetting the missing clock detect circuit and then re-locking the PLL. Post a re-lock the clock frequency would be 200Mhz but using the INTOSC1 as clock source.

External Connections

- None.

Watch Variables

- *fail* - Indicates that a missing clock was either not detected or was not handled correctly.
- *mcd_clkfail_isr* - Indicates that the missing clock failure caused an NMI to be triggered and called an the ISR to handle it.
- *mcd_detect* - Indicates that a missing clock was detected.
- *result* - Status of a successful handling of missing clock detection

3.12.8.2 XCLKOUT (External Clock Output) Configuration - SINGLE_CORE

FILE: sysctl_ex2_xclkout.c

This example demonstrates how to configure the XCLKOUT pin for observing internal clocks through an external pin, for debugging and testing purposes.

In this example, we are using INTOSC1 as the XCLKOUT clock source and configuring the divider as 8. Expected frequency of XCLKOUT = (INTOSC1 freq)/8 = 10/8 = 1.25MHz

View the XCLKOUT on GPIO73 using an oscilloscope.

3.12.9 TIMER Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/timer

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

3.12.9.1 Timer Academy Lab - SINGLE_CORE

FILE: timer_academy_lab.c

This example demonstrates how to use a CPU Timer to blink two LEDs. The timer and the LED GPIOs are each configured wthin the SysConfig file. The LED GPIO pins are toggled in the CPU Timer interrupt service routine. A global variable is incremented at the end of each timer interrupt.

External Connections

- None.

Watch Variables

- *cpuTimer0IntCount* - CPU Timer interrupt counts.

3.12.9.2 CPU Timers - SINGLE_CORE

FILE: timer_ex1_cputimers.c

This example configures CPU Timer0, 1, and 2 and increments a counter each time the timer asserts an interrupt.

External Connections

- None

Watch Variables

- cpuTimer0IntCount
- cpuTimer1IntCount
- cpuTimer2IntCount

3.12.9.3 CPU Timers - SINGLE_CORE

FILE: timer_ex2_cputimers_syscfg.c

This example configures CPU Timer0, 1, and 2 and increments a counter each time the timer asserts an interrupt.

The interrupt priorities are configured as follows :
- RTINT Threshold = 15
- Timer 0 Interrupt priority = 10 -> RTINT
- Timer 1 Interrupt priority = 20 -> INT
- Timer 2 Interrupt priority = 30 -> INT

Sysconfig inserts the required attributes to the ISR functions to inform the compiler that the function is an interrupt / realtime interrupt.

External Connections
- None

Watch Variables
- cpuTimer0IntCount
- cpuTimer1IntCount
- cpuTimer2IntCount

3.12.10 WATCHDOG Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/watchdog

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

3.12.10.1 Watchdog - SINGLE_CORE

FILE: watchdog_ex1_service.c

This example shows how to service the watchdog or generate a watchdog interrupt using the watchdog. By default the example will generate a watchdog interrupt. To service the watchdog and not generate the interrupt, uncomment the SysCtl_serviceWatchdog() line in the main for loop.

External Connections
- None.

Watch Variables
- ISRCount - The number of times entered into the watchdog ISR
- loopCount - The number of loops performed while not in ISR

3.12.11 LPM Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/lpm

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

3.12.11.1 Low Power Modes: Device Idle Mode and Wakeup using GPIO - SINGLE_CORE

FILE: lpm_ex1_idlewake_gpio.c

This example puts the device into IDLE mode and then wakes up the device from IDLE using XINT1 which triggers on a falling edge of GPIO0.

The GPIO0 pin must be pulled from high to low by an external agent for wakeup. GPIO0 is configured as an XINT1 pin to trigger an XINT1 interrupt upon detection of a falling edge.

Initially, pull GPIO0 high externally. To wake device from IDLE mode by triggering an XINT1 interrupt, pull GPIO0 low (falling edge). The wakeup process begins as soon as GPIO0 is held low for the time indicated in the device datasheet.

GPIO1 is pulled high before entering the IDLE mode and is pulled low when in the external interrupt ISR.

External Connections

- GPIO0 needs to be pulled low to wake up the device.
- On device wakeup, the GPIO1 will be low and LED1 will start blinking

3.12.11.2 Low Power Modes: Device Idle Mode and Wakeup using Watchdog - SINGLE_CORE

FILE: lpm_ex2_idlewake_watchdog.c

This example puts the device into IDLE mode and then wakes up the device from IDLE using watchdog timer.

The device wakes up from the IDLE mode when the watchdog timer overflows, triggering an interrupt. A pre scalar is set for the watchdog timer to change the counter overflow time.

GPIO1 is pulled high before entering the IDLE mode and is pulled low when in the wakeup ISR.

External Connections

- On device wakeup, the GPIO1 will be low and LED1 will start blinking

3.12.11.3 Low Power Modes: Device Standby Mode and Wakeup using GPIO - SINGLE_CORE

FILE: lpm_ex3_standbywake_gpio.c

This example puts the device into STANDBY mode. If the lowest possible current consumption in STANDBY mode is desired, the JTAG connector must be removed from the device board while the device is in STANDBY mode.

This example puts the device into STANDBY mode and then wakes up the device from STANDBY using an LPM wakeup pin.

The pin GPIO0 is configured as the LPM wakeup pin to trigger a WAKEINT interrupt upon detection of a low pulse. Initially, pull GPIO0 high externally. To wake device from STANDBY mode, pull GPIO0 low for at least (2+QUALSTDBY), OSCLKS, then pull it high again.

The example then wakes up the device from STANDBY using GPIO0. GPIO0 wakes the device from STANDBY mode when a low pulse (signal goes high->low->high)is detected on the pin. This pin must be pulsed by an external agent for wakeup.

GPIO1 is pulled high before entering the STANDBY mode and is pulled low when in the wakeup ISR.

External Connections

- GPIO0 needs to be pulled low to wake up the device.
- On device wakeup, the GPIO1 will be low and LED1 will start blinking

3.12.11.4 Low Power Modes: Device Standby Mode and Wakeup using Watchdog - SINGLE_CORE

FILE: lpm_ex4_standbywake_watchdog.c

This example puts the device into STANDBY mode. If the lowest possible current consumption in STANDBY mode is desired, the JTAG connector must be removed from the device board while the device is in STANDBY mode.

This example puts the device into STANDBY mode then wakes up the device from STANDBY using watchdog timer.

The device wakes up from the STANDBY mode when the watchdog timer overflows triggering an interrupt. In the ISR, the GPIO1 is pulled low. the GPIO1 is toggled to indicate the device is out of STANDBY mode. A pre scalar is set for the watchdog timer to change the counter overflow time.

GPIO1 is pulled high before entering the STANDBY mode and is pulled low when in the wakeup ISR.

External Connections

- On device wakeup, the GPIO1 will be low and LED1 will start blinking

3.13 SYSCTRL Registers

This Section describes the SYSCTRL Registers.

3.13.1 SYSCTRL Base Address Table

Table 3-19. SYSCTRL Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
DEV_CFG_REGS [1]	DEVCFG_BASE	0x3018_0000	-	YES	YES	YES	-	-	YES	YES
MEMSS_L_CONFIG_REGS [1]	MEMSSLCFG_BASE	0x301D_8000	-	YES	YES	YES	-	-	YES	YES
MEMSS_C_CONFIG_REGS [1]	MEMSSCCFG_BASE	0x301D_8400	-	YES	YES	YES	-	-	YES	YES
MEMSS_M_CONFIG_REGS [1]	MEMSSMCFG_BASE	0x301D_8800	-	YES	YES	YES	-	-	YES	YES
MEMSS_MISCI_REGS [1]	MEMSSMISCI_BASE	0x301D_8E00	-	YES	YES	YES	-	-	YES	YES
CPU_SYS_REGS	CPUSYS_BASE	0x3020_0000	-	YES	YES	YES	-	-	-	YES
CPU_PER_CFG_REGS	CPUPERCFG_BASE	0x3020_8000	-	YES	YES	YES	-	-	-	YES
WD_REGS	WD_BASE	0x3020_8C00	-	YES	YES	YES	-	-	-	YES
CPUTIMER_REGS	CPUTIMER0_BASE	0x3021_8000	-	YES	YES	YES	-	-	-	YES
CPUTIMER_REGS	CPUTIMER1_BASE	0x3021_9000	-	YES	YES	YES	-	-	-	YES
CPUTIMER_REGS	CPUTIMER2_BASE	0x3021_A000	-	YES	YES	YES	-	-	-	YES
XINT_REGS	XINT_BASE	0x3027_0000	-	YES	YES	YES	-	-	-	YES

(1) Registers writeable by CPU1.LINK0, CPU1.LINK1, CPU1.LINK2 only. All CPUs can read all registers in all LINKs. Debug write access only allowed if Zone0 or Zone1 are enabled for full debug by all CPUs. Debug reads always allowed. Register Read/Write access by HSM.

3.13.2 DEV_CFG_REGS Registers

Table 3-20 lists the memory-mapped registers for the DEV_CFG_REGS registers. All register offset addresses not listed in Table 3-20 should be considered as reserved locations and the register contents should not be modified.

Table 3-20. DEV_CFG_REGS Registers

Offset	Acronym	Register Name	Protection
0h	DEVCFGLOCK1	Lock bit for PERxSYSCONFIG0-31 registers	PARITY
4h	DEVCFGLOCK2	Lock bit for DEVCFG registers	PARITY
8h	DEVCFGLOCK3	Lock bit for PERxSYSCONFIG32-63 registers	PARITY
Ch	DEVCFGLOCK4	Lock bit for PERxSYSCONFIG64-95 registers	PARITY
10h	DEVCFGLOCK5	Lock bit for PERxSYSCONFIG96-127 registers	PARITY
14h	DEVCFGLOCK6	Lock bit for PERxSYSCONFIG128-159 registers	PARITY
20h	PARTIDL	Lower 32-bit of Device PART Identification Number	PARITY
24h	PARTIDH	Upper 32-bit of Device PART Identification Number	PARITY
28h	REVID	Device Revision Number	PARITY
1C0h	MCUCNF1	MCUCNF Capability: EMIF Customization	PARITY
1C4h	MCUCNF2	MCUCNF Capability: EPWM	PARITY
1CCh	MCUCNF4	MCUCNF Capability: EQEP	PARITY
1D8h	MCUCNF7	MCUCNF Capability: UART	PARITY
1E4h	MCUCNF10	MCUCNF Capability: CAN, MCAN	PARITY
1F0h	MCUCNF13	MCUCNF Capability: AMCUCNF	PARITY
1F4h	MCUCNF14	MCUCNF Capability: CMPSS	PARITY
1FCh	MCUCNF16	MCUCNF Capability: DAC	PARITY
200h	MCUCNF17	MCUCNF Capability: CLB	PARITY
204h	MCUCNF18	MCUCNF Capability: FSI	PARITY
208h	MCUCNF19	MCUCNF Capability: LIN	PARITY
218h	MCUCNF23	MCUCNF Capability: EtherCAT	PARITY
224h	MCUCNF26	Device Capability: HSM-Crypto Engines AES, SHA, PKA, TRNG	PARITY
238h	MCUCNF31	Device Capability: HSM-Crypto Engines SM2, SM3, SM4	PARITY
2BCh	MCUCNF64	MCUCNF Capability: MCUCNF level, Processing Block, RTDMA Customization	PARITY
2C0h	MCUCNF65	MCUCNF Capability: On-chip SRAM Customization	PARITY
2E4h	MCUCNF74	MCUCNF Capability: FLC1.B0/B1	PARITY
2ECh	MCUCNF76	MCUCNF Capability: FLC1.B2/B3	PARITY
2F4h	MCUCNF78	MCUCNF Capability: FLC1.B4 256KB Data Flash	PARITY
2F8h	MCUCNF79	MCUCNF Capability: FLC2.B0/B1	PARITY
300h	MCUCNF81	MCUCNF Capability: FLC2.B2/B3	PARITY
33Ch	MCUCNFLOCK1	Lock bit for MCUCNFx registers	PARITY
340h	MCUCNFLOCK2	Lock bit for MCUCNFx registers	PARITY
344h	MCUCNFLOCK3	Lock bit for MCUCNFx registers	PARITY
348h	LSEN	Lockstep enable configuration	PARITY
37Ch	EPWMXLINKCFG	Configure which EPWM module instaces are linked in the XLINK scheme	PARITY
384h	SICCONFIG	Safety Interconnect(SIC) Configuration - Enable and READY TIMEOUT value	
3B0h	RSTSTAT	Reset Status register for secondary CPUs	PARITY
3B4h	LPMSTAT	LPM Status Register for secondary CPUs	PARITY

Table 3-20. DEV_CFG_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
3CCh	TAP_STATUS	Status of JTAG State machine & Debugger Connect	PARITY
3D0h	TAP_CONTROL	Disable TAP control	PARITY
3D4h	DEVLIFECYCLE	Reflect the state of the Device Life Cycle signals reported from the HSM	PARITY
47Ch	SDFMTYPE	Configures SDFM Type for the device	PARITY
4ACh	SYNCSELECT	Sync Input and Output Select Register	PARITY
4B0h	ADCSOCOUTSELECT	External ADCSOC Select Register (PWM1-16)	PARITY
4B4h	ADCSOCOUTSELECT1	External ADCSOC Select Register (PWM17-32)	PARITY
4B8h	SYNCSOCLOCK	SYNCSEL and ADCSOC Select Lock register	PARITY
4DCh	HSMTOCPU_STS1	HSM to C29x Signal Status1	PARITY
4E0h	HSMTOCPU_STS2	HSM to C29x Signal Status2	PARITY
4E4h	HSM_SECURE_BOOT_INFO_REG0	Status information of the secure boot process HSM to CPU1	PARITY
4E8h	HSM_SECURE_BOOT_INFO_REG1	Status information of the secure boot process HSM to CPU1	PARITY
4ECh	HSM_SECURE_BOOT_INFO_REG2	Status information of the secure boot process HSM to CPU1	PARITY
4F0h	HSM_SECURE_BOOT_INFO_REG3	Status information of the secure boot process HSM to CPU1	PARITY
4F4h	HSM_SECURE_BOOT_INFO_REG4	Status information of the secure boot process HSM to CPU1	PARITY
4F8h	HSM_SECURE_BOOT_INFO_REG5	Status information of the secure boot process HSM to CPU1	PARITY
4FCh	HSM_SECURE_BOOT_INFO_REG6	Status information of the secure boot process HSM to CPU1	PARITY
500h	HSM_SECURE_BOOT_INFO_REG7	Status information of the secure boot process HSM to CPU1	PARITY
504h	SOC_SECURE_BOOT_INFO_REG0	Status information of the secure boot process CPU1 to HSM	PARITY
508h	SOC_SECURE_BOOT_INFO_REG1	Status information of the secure boot process CPU1 to HSM	PARITY
50Ch	SOC_SECURE_BOOT_INFO_REG2	Status information of the secure boot process CPU1 to HSM	PARITY
510h	SOC_SECURE_BOOT_INFO_REG3	Status information of the secure boot process CPU1 to HSM	PARITY
514h	SOC_SECURE_BOOT_INFO_REG4	Status information of the secure boot process CPU1 to HSM	PARITY
518h	SOC_SECURE_BOOT_INFO_REG5	Status information of the secure boot process CPU1 to HSM	PARITY
51Ch	SOC_SECURE_BOOT_INFO_REG6	Status information of the secure boot process CPU1 to HSM	PARITY
520h	SOC_SECURE_BOOT_INFO_REG7	Status information of the secure boot process CPU1 to HSM	PARITY
524h	CLKCFGLOCK1	Lock bit for CLKCFG registers	PARITY
530h	CLKSRCCTL1	Clock Source Control register-1	PARITY
534h	CLKSRCCTL2	Clock Source Control register-2	PARITY
538h	CLKSRCCTL3	Clock Source Control register-3	PARITY
53Ch	SYSPLLCTL1	SYSPLL Control register-1	PARITY
548h	SYSPLLMULT	SYSPLL Multiplier register	PARITY
54Ch	SYSPLLSTS	SYSPLL Status register	PARITY

Table 3-20. DEV_CFG_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
564h	SYSCLKDIVSEL	System Clock Divider Select register	PARITY
56Ch	PERCLKDIVSEL	Peripheral Clock Divider Select register	PARITY
570h	XCLKOUTDIVSEL	XCLKOUT Divider Select register	PARITY
574h	HSMCLKDIVSEL	HSM SYSCLK Divider Select register	PARITY
578h	MCANCLKDIVSEL	MCAN Bit Clock Divider Select register	PARITY
57Ch	CLBCLKCTL	CLB Clocking Control Register	PARITY
584h	MCDCR	Missing Clock Detect Control Register	PARITY
588h	X1CNT	10-bit Counter on X1 Clock	
58Ch	XTALCR	XTAL Control Register	PARITY
59Ch	XTALCR2	XTAL Control Register for pad init	PARITY
5A8h	ETHERCATCLKCTL	EtherCAT Clock Control	PARITY
5ACh	ETHERCATCTL	ETHERCAT control register.	PARITY
5B0h	SYNCBUSY	Pulse Transfer Sync Busy Status register	PARITY
5C0h	ESMXRSNCTL	Enable ESM reset outputs for XRSn	PARITY
5C8h	EPWM1	PER2SYSCONFIG - Peripheral System Configuration for EPWM1	PARITY
5CCh	EPWM2	PER3SYSCONFIG - Peripheral System Configuration for EPWM2	PARITY
5D0h	EPWM3	PER4SYSCONFIG - Peripheral System Configuration for EPWM3	PARITY
5D4h	EPWM4	PER5SYSCONFIG - Peripheral System Configuration for EPWM4	PARITY
5D8h	EPWM5	PER6SYSCONFIG - Peripheral System Configuration for EPWM5	PARITY
5DCh	EPWM6	PER7SYSCONFIG - Peripheral System Configuration for EPWM6	PARITY
5E0h	EPWM7	PER8SYSCONFIG - Peripheral System Configuration for EPWM7	PARITY
5E4h	EPWM8	PER9SYSCONFIG - Peripheral System Configuration for EPWM8	PARITY
5E8h	EPWM9	PER10SYSCONFIG - Peripheral System Configuration for EPWM9	PARITY
5ECh	EPWM10	PER11SYSCONFIG - Peripheral System Configuration for EPWM10	PARITY
5F0h	EPWM11	PER12SYSCONFIG - Peripheral System Configuration for EPWM11	PARITY
5F4h	EPWM12	PER13SYSCONFIG - Peripheral System Configuration for EPWM12	PARITY
5F8h	EPWM13	PER14SYSCONFIG - Peripheral System Configuration for EPWM13	PARITY
5FCh	EPWM14	PER15SYSCONFIG - Peripheral System Configuration for EPWM14	PARITY
600h	EPWM15	PER16SYSCONFIG - Peripheral System Configuration for EPWM15	PARITY
604h	EPWM16	PER17SYSCONFIG - Peripheral System Configuration for EPWM16	PARITY
608h	EPWM17	PER18SYSCONFIG - Peripheral System Configuration for EPWM17	PARITY
60Ch	EPWM18	PER19SYSCONFIG - Peripheral System Configuration for EPWM18	PARITY

Table 3-20. DEV_CFG_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
614h	HRCAL0	PER21SYSCONFIG - Peripheral System Configuration for HRCAL0	PARITY
618h	HRCAL1	PER22SYSCONFIG - Peripheral System Configuration for HRCAL1	PARITY
61Ch	HRCAL2	PER23SYSCONFIG - Peripheral System Configuration for HRCAL2	PARITY
620h	ECAP1	PER24SYSCONFIG - Peripheral System Configuration for ECAP1	PARITY
624h	ECAP2	PER25SYSCONFIG - Peripheral System Configuration for ECAP2	PARITY
628h	ECAP3	PER26SYSCONFIG - Peripheral System Configuration for ECAP3	PARITY
62Ch	ECAP4	PER27SYSCONFIG - Peripheral System Configuration for ECAP4	PARITY
630h	ECAP5	PER28SYSCONFIG - Peripheral System Configuration for ECAP5	PARITY
634h	ECAP6	PER29SYSCONFIG - Peripheral System Configuration for ECAP6	PARITY
638h	EQEP1	PER30SYSCONFIG - Peripheral System Configuration for EQEP1	PARITY
63Ch	EQEP2	PER31SYSCONFIG - Peripheral System Configuration for EQEP2	PARITY
640h	EQEP3	PER32SYSCONFIG - Peripheral System Configuration for EQEP3	PARITY
644h	EQEP4	PER33SYSCONFIG - Peripheral System Configuration for EQEP4	PARITY
648h	EQEP5	PER34SYSCONFIG - Peripheral System Configuration for EQEP5	PARITY
64Ch	EQEP6	PER35SYSCONFIG - Peripheral System Configuration for EQEP6	PARITY
650h	SDFM1	PER36SYSCONFIG - Peripheral System Configuration for SDFM1	PARITY
654h	SDFM2	PER37SYSCONFIG - Peripheral System Configuration for SDFM2	PARITY
658h	SDFM3	PER38SYSCONFIG - Peripheral System Configuration for SDFM3	PARITY
65Ch	SDFM4	PER39SYSCONFIG - Peripheral System Configuration for SDFM4	PARITY
660h	UARTA	PER40SYSCONFIG - Peripheral System Configuration for UARTA	PARITY
664h	UARTB	PER41SYSCONFIG - Peripheral System Configuration for UARTB	PARITY
668h	UARTC	PER42SYSCONFIG - Peripheral System Configuration for UARTC	PARITY
66Ch	UARTD	PER43SYSCONFIG - Peripheral System Configuration for UARTD	PARITY
670h	UARTE	PER44SYSCONFIG - Peripheral System Configuration for UARTE	PARITY
674h	UARTF	PER45SYSCONFIG - Peripheral System Configuration for UARTF	PARITY
678h	SPIA	PER46SYSCONFIG - Peripheral System Configuration for SPIA	PARITY

Table 3-20. DEV_CFG_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
67Ch	SPIB	PER47SYSCONFIG - Peripheral System Configuration for SPIB	PARITY
680h	SPIC	PER48SYSCONFIG - Peripheral System Configuration for SPIC	PARITY
684h	SPID	PER49SYSCONFIG - Peripheral System Configuration for SPID	PARITY
688h	SPIE	PER50SYSCONFIG - Peripheral System Configuration for SPIE	PARITY
68Ch	I2CA	PER51SYSCONFIG - Peripheral System Configuration for I2CA	PARITY
690h	I2CB	PER52SYSCONFIG - Peripheral System Configuration for I2CB	PARITY
694h	PMBUSA	PER53SYSCONFIG - Peripheral System Configuration for PMBUSA	PARITY
698h	LINA	PER54SYSCONFIG - Peripheral System Configuration for LINA	PARITY
69Ch	LINB	PER55SYSCONFIG - Peripheral System Configuration for LINB	PARITY
6A0h	MCANA	PER56SYSCONFIG - Peripheral System Configuration for MCANA	PARITY
6A4h	MCANB	PER57SYSCONFIG - Peripheral System Configuration for MCANB	PARITY
6A8h	MCANC	PER58SYSCONFIG - Peripheral System Configuration for MCANC	PARITY
6ACh	MCAND	PER59SYSCONFIG - Peripheral System Configuration for MCAND	PARITY
6B0h	MCANE	PER60SYSCONFIG - Peripheral System Configuration for MCANE	PARITY
6B4h	MCANF	PER61SYSCONFIG - Peripheral System Configuration for MCANF	PARITY
6B8h	ADCA	PER62SYSCONFIG - Peripheral System Configuration for ADCA	PARITY
6BCh	ADCB	PER63SYSCONFIG - Peripheral System Configuration for ADCB	PARITY
6C0h	ADCC	PER64SYSCONFIG - Peripheral System Configuration for ADCC	PARITY
6C4h	ADCD	PER65SYSCONFIG - Peripheral System Configuration for ADCD	PARITY
6C8h	ADCE	PER66SYSCONFIG - Peripheral System Configuration for ADCE	PARITY
6CCh	CMPSS1	PER67SYSCONFIG - Peripheral System Configuration for CMPSS1	PARITY
6D0h	CMPSS2	PER68SYSCONFIG - Peripheral System Configuration for CMPSS2	PARITY
6D4h	CMPSS3	PER69SYSCONFIG - Peripheral System Configuration for CMPSS3	PARITY
6D8h	CMPSS4	PER70SYSCONFIG - Peripheral System Configuration for CMPSS4	PARITY
6DCh	CMPSS5	PER71SYSCONFIG - Peripheral System Configuration for CMPSS5	PARITY
6E0h	CMPSS6	PER72SYSCONFIG - Peripheral System Configuration for CMPSS6	PARITY

Table 3-20. DEV_CFG_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
6E4h	CMPSS7	PER73SYSCONFIG - Peripheral System Configuration for CMPSS7	PARITY
6E8h	CMPSS8	PER74SYSCONFIG - Peripheral System Configuration for CMPSS8	PARITY
6ECh	CMPSS9	PER75SYSCONFIG - Peripheral System Configuration for CMPSS9	PARITY
6F0h	CMPSS10	PER76SYSCONFIG - Peripheral System Configuration for CMPSS10	PARITY
6F4h	CMPSS11	PER77SYSCONFIG - Peripheral System Configuration for CMPSS11	PARITY
6F8h	CMPSS12	PER78SYSCONFIG - Peripheral System Configuration for CMPSS12	PARITY
6FCh	DACA	PER79SYSCONFIG - Peripheral System Configuration for DACA	PARITY
700h	DACB	PER80SYSCONFIG - Peripheral System Configuration for DACB	PARITY
704h	CLB1	PER81SYSCONFIG - Peripheral System Configuration for CLB1	PARITY
708h	CLB2	PER82SYSCONFIG - Peripheral System Configuration for CLB2	PARITY
70Ch	CLB3	PER83SYSCONFIG - Peripheral System Configuration for CLB3	PARITY
710h	CLB4	PER84SYSCONFIG - Peripheral System Configuration for CLB4	PARITY
714h	CLB5	PER85SYSCONFIG - Peripheral System Configuration for CLB5	PARITY
718h	CLB6	PER86SYSCONFIG - Peripheral System Configuration for CLB6	PARITY
71Ch	FSITXA	PER87SYSCONFIG - Peripheral System Configuration for FSITXA	PARITY
720h	FSITXB	PER88SYSCONFIG - Peripheral System Configuration for FSITXB	PARITY
724h	FSITXC	PER89SYSCONFIG - Peripheral System Configuration for FSITXC	PARITY
728h	FSITXD	PER90SYSCONFIG - Peripheral System Configuration for FSITXD	PARITY
72Ch	FSIRXA	PER91SYSCONFIG - Peripheral System Configuration for FSIRXA	PARITY
730h	FSIRXB	PER92SYSCONFIG - Peripheral System Configuration for FSIRXB	PARITY
734h	FSIRXC	PER93SYSCONFIG - Peripheral System Configuration for FSIRXC	PARITY
738h	FSIRXD	PER94SYSCONFIG - Peripheral System Configuration for FSIRXD	PARITY
73Ch	DCC1	PER95SYSCONFIG - Peripheral System Configuration for DCC1	PARITY
740h	DCC2	PER96SYSCONFIG - Peripheral System Configuration for DCC2	PARITY
744h	DCC3	PER97SYSCONFIG - Peripheral System Configuration for DCC3	PARITY
748h	ETHERCATA	PER98SYSCONFIG - Peripheral System Configuration for ETHERCATA	PARITY

Table 3-20. DEV_CFG_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
74Ch	EPG1	PER99SYSCONFIG - Peripheral System Configuration for EPG1	PARITY
750h	SENT1	PER100SYSCONFIG - Peripheral System Configuration for SENT1	PARITY
754h	SENT2	PER101SYSCONFIG - Peripheral System Configuration for SENT2	PARITY
758h	SENT3	PER102SYSCONFIG - Peripheral System Configuration for SENT3	PARITY
75Ch	SENT4	PER103SYSCONFIG - Peripheral System Configuration for SENT4	PARITY
760h	SENT5	PER104SYSCONFIG - Peripheral System Configuration for SENT5	PARITY
764h	SENT6	PER105SYSCONFIG - Peripheral System Configuration for SENT6	PARITY
768h	ADCCHECKER1	PER106SYSCONFIG - Peripheral System Configuration for ADCCHECKER1	PARITY
76Ch	ADCCHECKER2	PER107SYSCONFIG - Peripheral System Configuration for ADCCHECKER2	PARITY
770h	ADCCHECKER3	PER108SYSCONFIG - Peripheral System Configuration for ADCCHECKER3	PARITY
774h	ADCCHECKER4	PER109SYSCONFIG - Peripheral System Configuration for ADCCHECKER4	PARITY
778h	ADCCHECKER5	PER110SYSCONFIG - Peripheral System Configuration for ADCCHECKER5	PARITY
77Ch	ADCCHECKER6	PER111SYSCONFIG - Peripheral System Configuration for ADCCHECKER6	PARITY
780h	ADCCHECKER7	PER112SYSCONFIG - Peripheral System Configuration for ADCCHECKER7	PARITY
784h	ADCCHECKER8	PER113SYSCONFIG - Peripheral System Configuration for ADCCHECKER8	PARITY
788h	ADCCHECKER9	PER114SYSCONFIG - Peripheral System Configuration for ADCCHECKER9	PARITY
78Ch	ADCCHECKER10	PER115SYSCONFIG - Peripheral System Configuration for ADCCHECKER10	PARITY
790h	ADCSEAGGRCPU1	PER116SYSCONFIG - Peripheral System Configuration for ADCSEAGGRCPU1	PARITY
794h	ADCSEAGGRCPU2	PER117SYSCONFIG - Peripheral System Configuration for ADCSEAGGRCPU2	PARITY
798h	ADCSEAGGRCPU3	PER118SYSCONFIG - Peripheral System Configuration for ADCSEAGGRCPU3	PARITY
7A8h	RTDMA1CH	PER122SYSCONFIG - Peripheral System Configuration for RTDMA1CH	PARITY
7ACh	RTDMA2CH	PER123SYSCONFIG - Peripheral System Configuration for RTDMA2CH	PARITY
7B0h	WADI1	PER124SYSCONFIG - Peripheral System Configuration for WADI1	PARITY
7B4h	WADI2	PER125SYSCONFIG - Peripheral System Configuration for WADI2	PARITY
7B8h	INPUTXBARFlags	PER126SYSCONFIG - Peripheral System Configuration for INPUTXBARFlags	PARITY
7BCh	OUTPUTXBARFlags	PER127SYSCONFIG - Peripheral System Configuration for OUTPUTXBARFlags	PARITY

Table 3-20. DEV_CFG_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
7C0h	DLTFIFORegs	PER128SYSCONFIG - Peripheral System Configuration for DLTFIFORegs	PARITY
7C4h	ADC_GLOBAL_REGS	PER129SYSCONFIG - Peripheral System Configuration for ADC_GLOBAL_REGS	PARITY
7C8h	Error_Aggregator	PER130SYSCONFIG - Peripheral System Configuration for Error_Aggregator	PARITY
7CCh	ESM	PER131SYSCONFIG - Peripheral System Configuration for ESM ESMCPU1/2/3 and ESMSYS	PARITY
7E4h	PARITY_TEST	Enables parity test	

Complex bit access types are encoded to fit into small table cells. Table 3-21 shows the codes that are used for access types in this section.

Table 3-21. DEV_CFG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
WOnce	W Once	Write Write once
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.13.2.1 DEVCFGLOCK1 Register (Offset = 0h) [Reset = 00000000h]

DEVCFGLOCK1 is shown in Figure 3-28 and described in Table 3-22.

Return to the Summary Table.

Lock bit for CPUSELx registers
The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed
Note:
Any SOnce bit in this register, once set can only be cleared through a CPU1.SYSRSn. Write of 0 to any bit of this register has no effect

Figure 3-28. DEVCFGLOCK1 Register

31	30	29	28	27	26	25	24
PER31SYSCONFIG	PER30SYSCONFIG	PER29SYSCONFIG	PER28SYSCONFIG	PER27SYSCONFIG	PER26SYSCONFIG	PER25SYSCONFIG	PER24SYSCONFIG
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
PER23SYSCONFIG	PER22SYSCONFIG	PER21SYSCONFIG	PER20SYSCONFIG	PER19SYSCONFIG	PER18SYSCONFIG	PER17SYSCONFIG	PER16SYSCONFIG
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
PER15SYSCONFIG	PER14SYSCONFIG	PER13SYSCONFIG	PER12SYSCONFIG	PER11SYSCONFIG	PER10SYSCONFIG	PER9SYSCONFIG	PER8SYSCONFIG
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
PER7SYSCONFIG	PER6SYSCONFIG	PER5SYSCONFIG	PER4SYSCONFIG	PER3SYSCONFIG	PER2SYSCONFIG	PER1SYSCONFIG	PER0SYSCONFIG
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 3-22. DEVCFGLOCK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PER31SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
30	PER30SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
29	PER29SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
28	PER28SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
27	PER27SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
26	PER26SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
25	PER25SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn

Table 3-22. DEVCFGLOCK1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	PER24SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
23	PER23SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
22	PER22SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
21	PER21SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
20	PER20SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
19	PER19SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
18	PER18SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
17	PER17SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
16	PER16SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
15	PER15SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
14	PER14SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
13	PER13SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
12	PER12SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
11	PER11SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
10	PER10SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
9	PER9SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
8	PER8SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn

Table 3-22. DEVCFGLOCK1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	PER7SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
6	PER6SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
5	PER5SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
4	PER4SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
3	PER3SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
2	PER2SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
1	PER1SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
0	PER0SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn

3.13.2.2 DEVCFGLOCK2 Register (Offset = 4h) [Reset = 00000000h]

DEVCFGLOCK2 is shown in Figure 3-29 and described in Table 3-23.

Return to the Summary Table.

Lock bit for DEVCFG registers
Note:
[1] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed
[2] Any SOnce bit in this register, once set can only be cleared through a CPU1.SYSRSn. Write of 0 to any bit of this register has no effect

Figure 3-29. DEVCFGLOCK2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
PCLKCR22	ETHERCATCTL	RESERVED	LSEN	SICCONFIG	RESERVED	RESERVED	RESERVED
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 3-23. DEVCFGLOCK2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	PCLKCR22	R/WSonce	0h	0 Allows write to PCLKCR22 register 1 Blocks write to PCLKCR22 register Reset type: CPU1.SYSRSn
6	ETHERCATCTL	R/WSonce	0h	0 Allows write to ETHERCATCTL register 1 Blocks write to ETHERCATCTL register Reset type: CPU1.SYSRSn
5	RESERVED	R/WSonce	0h	Reserved
4	LSEN	R/WSonce	0h	0 Allows write to LSEN register 1 Blocks write to LSEN register Reset type: CPU1.SYSRSn
3	SICCONFIG	R/WSonce	0h	0 Allows write to SICCONFIG register 1 Blocks write to SICCONFIG register Reset type: CPU1.SYSRSn
2	RESERVED	R/WSonce	0h	Reserved
1	RESERVED	R/WSonce	0h	Reserved
0	RESERVED	R/WSonce	0h	Reserved

3.13.2.3 DEVCFGLOCK3 Register (Offset = 8h) [Reset = 00000000h]

DEVCFGLOCK3 is shown in Figure 3-30 and described in Table 3-24.

Return to the Summary Table.

Lock bit for DEVCFG registers
Note:
[1] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed
[2] Any SOnce bit in this register, once set can only be cleared through a CPU1.SYSRSn. Write of 0 to any bit of this register has no effect

Figure 3-30. DEVCFGLOCK3 Register

31	30	29	28	27	26	25	24
PER63SYSCONFIG	PER62SYSCONFIG	PER61SYSCONFIG	PER60SYSCONFIG	PER59SYSCONFIG	PER58SYSCONFIG	PER57SYSCONFIG	PER56SYSCONFIG
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
PER55SYSCONFIG	PER54SYSCONFIG	PER53SYSCONFIG	PER52SYSCONFIG	PER51SYSCONFIG	PER50SYSCONFIG	PER49SYSCONFIG	PER48SYSCONFIG
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
PER47SYSCONFIG	PER46SYSCONFIG	PER45SYSCONFIG	PER44SYSCONFIG	PER43SYSCONFIG	PER42SYSCONFIG	PER41SYSCONFIG	PER40SYSCONFIG
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
PER39SYSCONFIG	PER38SYSCONFIG	PER37SYSCONFIG	PER36SYSCONFIG	PER35SYSCONFIG	PER34SYSCONFIG	PER33SYSCONFIG	PER32SYSCONFIG
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 3-24. DEVCFGLOCK3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PER63SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
30	PER62SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
29	PER61SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
28	PER60SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
27	PER59SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
26	PER58SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
25	PER57SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn

Table 3-24. DEVCFGLOCK3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	PER56SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
23	PER55SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
22	PER54SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
21	PER53SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
20	PER52SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
19	PER51SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
18	PER50SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
17	PER49SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
16	PER48SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
15	PER47SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
14	PER46SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
13	PER45SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
12	PER44SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
11	PER43SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
10	PER42SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
9	PER41SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
8	PER40SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn

Table 3-24. DEVCFGLOCK3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	PER39SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
6	PER38SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
5	PER37SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
4	PER36SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
3	PER35SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
2	PER34SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
1	PER33SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
0	PER32SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn

3.13.2.4 DEVCFGLOCK4 Register (Offset = Ch) [Reset = 00000000h]

DEVCFGLOCK4 is shown in Figure 3-31 and described in Table 3-25.

Return to the Summary Table.

Lock bit for DEVCFG registers
Note:
[1] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed
[2] Any SOnce bit in this register, once set can only be cleared through a CPU1.SYSRSn. Write of 0 to any bit of this register has no effect

Figure 3-31. DEVCFGLOCK4 Register

31	30	29	28	27	26	25	24
PER95SYSCONFIG	PER94SYSCONFIG	PER93SYSCONFIG	PER92SYSCONFIG	PER91SYSCONFIG	PER90SYSCONFIG	PER89SYSCONFIG	PER88SYSCONFIG
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
PER87SYSCONFIG	PER86SYSCONFIG	PER85SYSCONFIG	PER84SYSCONFIG	PER83SYSCONFIG	PER82SYSCONFIG	PER81SYSCONFIG	PER80SYSCONFIG
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
PER79SYSCONFIG	PER78SYSCONFIG	PER77SYSCONFIG	PER76SYSCONFIG	PER75SYSCONFIG	PER74SYSCONFIG	PER73SYSCONFIG	PER72SYSCONFIG
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
PER71SYSCONFIG	PER70SYSCONFIG	PER69SYSCONFIG	PER68SYSCONFIG	PER67SYSCONFIG	PER66SYSCONFIG	PER65SYSCONFIG	PER64SYSCONFIG
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 3-25. DEVCFGLOCK4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PER95SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
30	PER94SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
29	PER93SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
28	PER92SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
27	PER91SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
26	PER90SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
25	PER89SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn

Table 3-25. DEVCFGLOCK4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	PER88SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
23	PER87SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
22	PER86SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
21	PER85SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
20	PER84SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
19	PER83SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
18	PER82SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
17	PER81SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
16	PER80SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
15	PER79SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
14	PER78SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
13	PER77SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
12	PER76SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
11	PER75SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
10	PER74SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
9	PER73SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
8	PER72SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn

Table 3-25. DEVCFGLOCK4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	PER71SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
6	PER70SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
5	PER69SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
4	PER68SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
3	PER67SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
2	PER66SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
1	PER65SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
0	PER64SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn

3.13.2.5 DEVCFGLOCK5 Register (Offset = 10h) [Reset = 00000000h]

DEVCFGLOCK5 is shown in Figure 3-32 and described in Table 3-26.

Return to the Summary Table.

Lock bit for DEVCFG registers
Note:
[1] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed
[2] Any SOnce bit in this register, once set can only be cleared through a CPU1.SYSRSn. Write of 0 to any bit of this register has no effect

Figure 3-32. DEVCFGLOCK5 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	PER120SYSCONFIG
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
PER119SYSCONFIG	PER118SYSCONFIG	PER117SYSCONFIG	PER116SYSCONFIG	PER115SYSCONFIG	PER114SYSCONFIG	PER113SYSCONFIG	PER112SYSCONFIG
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
PER111SYSCONFIG	PER110SYSCONFIG	PER109SYSCONFIG	PER108SYSCONFIG	PER107SYSCONFIG	PER106SYSCONFIG	PER105SYSCONFIG	PER104SYSCONFIG
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
PER103SYSCONFIG	PER102SYSCONFIG	PER101SYSCONFIG	PER100SYSCONFIG	PER99SYSCONFIG	PER98SYSCONFIG	PER97SYSCONFIG	PER96SYSCONFIG
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 3-26. DEVCFGLOCK5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/WSonce	0h	Reserved
30	RESERVED	R/WSonce	0h	Reserved
29	RESERVED	R/WSonce	0h	Reserved
28	RESERVED	R/WSonce	0h	Reserved
27	RESERVED	R/WSonce	0h	Reserved
26	RESERVED	R/WSonce	0h	Reserved
25	RESERVED	R/WSonce	0h	Reserved
24	PER120SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
23	PER119SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
22	PER118SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
21	PER117SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn

Table 3-26. DEVCFGLOCK5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	PER116SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
19	PER115SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
18	PER114SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
17	PER113SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
16	PER112SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
15	PER111SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
14	PER110SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
13	PER109SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
12	PER108SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
11	PER107SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
10	PER106SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
9	PER105SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
8	PER104SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
7	PER103SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
6	PER102SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
5	PER101SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
4	PER100SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn

Table 3-26. DEVCFGLOCK5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	PER99SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
2	PER98SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
1	PER97SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
0	PER96SYSCONFIG	R/WSonce	0h	0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn

3.13.2.6 DEVCFGLOCK6 Register (Offset = 14h) [Reset = 00000000h]

DEVCFGLOCK6 is shown in Figure 3-33 and described in Table 3-27.

Return to the Summary Table.

Lock bit for DEVCFG registers
Note:
[1] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed
[2] Any SOnce bit in this register, once set can only be cleared through a CPU1.SYSRSn. Write of 0 to any bit of this register has no effect

Figure 3-33. DEVCFGLOCK6 Register

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

Table 3-27. DEVCFGLOCK6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R-0	0h	Reserved
6	RESERVED	R/WSonce	0h	Reserved
5	RESERVED	R/WSonce	0h	Reserved
4	RESERVED	R/WSonce	0h	Reserved
3	RESERVED	R/WSonce	0h	Reserved
2	RESERVED	R/WSonce	0h	Reserved
1	RESERVED	R/WSonce	0h	Reserved
0	RESERVED	R/WSonce	0h	Reserved

3.13.2.7 PARTIDL Register (Offset = 20h) [Reset = 00XXXXX0h]

PARTIDL is shown in Figure 3-34 and described in Table 3-28.

Return to the Summary Table.

Lower 32-bit of Device PART Identification Number

Figure 3-34. PARTIDL Register

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

Table 3-28. PARTIDL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	PARTID_FORMAT_REV	R	0h	0 = Gen3 1 = F29x Devices Reset type: PORESETn
27-24	RESERVED	R	0h	Reserved
23-16	FLASH_SIZE	R	XXh	Flash Size 0x3=1MB 0x4=2MB/4 Banks (F29P58x) 0x5=2MB/8 Banks (F29H85x) 0x6=4MB Others=Reserved Reset type: PORESETn
15	RESERVED	R	0h	Reserved
14-13	RESERVED	R	Xh	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	Xh	Reserved
10-8	PIN_COUNT	R	Xh	0 = Reserved 1 = 100 pin QFP 2 = 144 pin QFP 3 = 176 pin QFP 4 = 256 pin BGA 5,6,7 = Reserved Reset type: PORESETn
7-6	QUAL	R	Xh	0 = Engineering sample (TMX) 1 = Pilot production (TMP) 2 = Fully qualified (TMS) Reset type: PORESETn
5	RESERVED	R	0h	Reserved
4-3	RESERVED	R	0h	Reserved
2-0	RESERVED	R	0h	Reserved

3.13.2.8 PARTIDH Register (Offset = 24h) [Reset = 0DXX0500h]

PARTIDH is shown in Figure 3-35 and described in Table 3-29.

Return to the Summary Table.

Upper 32-bit of Device PART Identification Number

Figure 3-35. PARTIDH Register

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

Table 3-29. PARTIDH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DEVICE_CLASS_ID	R	Dh	Device class ID Refer to the device specific datasheet for more information Reset type: PORESETn
23-16	PARTNO	R	XXh	Part Number Designator Refer to the device specific datasheet for more information Reset type: PORESETn
15-8	FAMILY	R	5h	Device Family This field categorizes the device to one of the C2000 device families. Reset type: PORESETn
7-4	RESERVED	R	0h	Reserved
3-0	RESERVED	R	0h	Reserved

3.13.2.9 REVID Register (Offset = 28h) [Reset = 00000000h]

REVID is shown in Figure 3-36 and described in Table 3-30.

Return to the Summary Table.

Device Revision Number

Figure 3-36. REVID Register

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

Table 3-30. REVID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	REVID	R/WOnce	0h	Device Revision ID. Loaded from flash trim sector by boot rom. Reset value is die-specific. Reset type: XRSn

3.13.2.10 MCUCNF1 Register (Offset = 1C0h) [Reset = 0000000Xh]

MCUCNF1 is shown in Figure 3-37 and described in Table 3-31.

Return to the Summary Table.

MCUCNF Capability: EMIF Customization

Figure 3-37. MCUCNF1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	EMIF1
R-0-0h						R-Xh	R-Xh

Table 3-31. MCUCNF1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	RESERVED	R	Xh	Reserved
0	EMIF1	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn

3.13.2.11 MCUCNF2 Register (Offset = 1C4h) [Reset = 000XXXXXh]

MCUCNF2 is shown in Figure 3-38 and described in Table 3-32.

Return to the Summary Table.

MCUCNF Capability: EPWM

Figure 3-38. MCUCNF2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						EPWM18	EPWM17
R-0-0h						R-Xh	R-Xh
15	14	13	12	11	10	9	8
EPWM16	EPWM15	EPWM14	EPWM13	EPWM12	EPWM11	EPWM10	EPWM9
R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh
7	6	5	4	3	2	1	0
EPWM8	EPWM7	EPWM6	EPWM5	EPWM4	EPWM3	EPWM2	EPWM1
R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh

Table 3-32. MCUCNF2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	Reserved
17	EPWM18	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
16	EPWM17	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
15	EPWM16	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
14	EPWM15	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
13	EPWM14	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
12	EPWM13	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
11	EPWM12	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
10	EPWM11	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
9	EPWM10	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn

Table 3-32. MCUCNF2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	EPWM9	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
7	EPWM8	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
6	EPWM7	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
5	EPWM6	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
4	EPWM5	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
3	EPWM4	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
2	EPWM3	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
1	EPWM2	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
0	EPWM1	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn

3.13.2.12 MCUCNF4 Register (Offset = 1CCh) [Reset = 000000XXh]

MCUCNF4 is shown in Figure 3-39 and described in Table 3-33.

Return to the Summary Table.

MCUCNF Capability: EQEP

Figure 3-39. MCUCNF4 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		EQEP6	EQEP5	EQEP4	EQEP3	EQEP2	EQEP1
R-0-0h		R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh

Table 3-33. MCUCNF4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	EQEP6	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
4	EQEP5	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
3	EQEP4	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
2	EQEP3	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
1	EQEP2	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
0	EQEP1	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn

3.13.2.13 MCUCNF7 Register (Offset = 1D8h) [Reset = 00XX001Xh]

MCUCNF7 is shown in Figure 3-40 and described in Table 3-34.

Return to the Summary Table.

MCUCNF Capability: UART

Figure 3-40. MCUCNF7 Register

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

Table 3-34. MCUCNF7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R-0	0h	Reserved
21	UART_F	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
20	UART_E	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
19	UART_D	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
18	UART_C	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
17	UART_B	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
16	UART_A	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
15-4	RESERVED	R-0	1h	Reserved
3	RESERVED	R	Xh	Reserved
2	RESERVED	R	Xh	Reserved
1	RESERVED	R	Xh	Reserved
0	RESERVED	R	Xh	Reserved

3.13.2.14 MCUCNF10 Register (Offset = 1E4h) [Reset = 00000XXXh]

MCUCNF10 is shown in Figure 3-41 and described in Table 3-35.

Return to the Summary Table.

MCUCNF Capability: CAN, MCAN

Figure 3-41. MCUCNF10 Register

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

Table 3-35. MCUCNF10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R-0	0h	Reserved
9	MCAN_F	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
8	MCAN_E	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
7	MCAN_D	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
6	MCAN_C	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
5	MCAN_B	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
4	MCAN_A	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
3	RESERVED	R	Xh	Reserved
2	RESERVED	R	Xh	Reserved
1	RESERVED	R	Xh	Reserved
0	RESERVED	R	Xh	Reserved

3.13.2.15 MCUCNF13 Register (Offset = 1F0h) [Reset = 0000001Fh]

MCUCNF13 is shown in Figure 3-42 and described in Table 3-36.

Return to the Summary Table.

MCUCNF Capability: AMCUCNF

Figure 3-42. MCUCNF13 Register

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

Table 3-36. MCUCNF13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	ADC_E	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
3	ADC_D	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
2	ADC_C	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
1	ADC_B	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
0	ADC_A	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn

3.13.2.16 MCUCNF14 Register (Offset = 1F4h) [Reset = 00000FFFh]

MCUCNF14 is shown in Figure 3-43 and described in Table 3-37.

Return to the Summary Table.

MCUCNF Capability: CMPSS

Figure 3-43. MCUCNF14 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				CMPSS12	CMPSS11	CMPSS10	CMPSS9
R-0-0h				R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
CMPSS8	CMPSS7	CMPSS6	CMPSS5	CMPSS4	CMPSS3	CMPSS2	CMPSS1
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 3-37. MCUCNF14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11	CMPSS12	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
10	CMPSS11	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
9	CMPSS10	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
8	CMPSS9	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
7	CMPSS8	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
6	CMPSS7	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
5	CMPSS6	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
4	CMPSS5	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
3	CMPSS4	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn

Table 3-37. MCUCNF14 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	CMPSS3	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
1	CMPSS2	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
0	CMPSS1	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn

3.13.2.17 MCUCNF16 Register (Offset = 1FCh) [Reset = 00030000h]

MCUCNF16 is shown in Figure 3-44 and described in Table 3-38.

Return to the Summary Table.

MCUCNF Capability: DAC

Figure 3-44. MCUCNF16 Register

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

Table 3-38. MCUCNF16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	DAC_B	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
16	DAC_A	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
15-4	RESERVED	R-0	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

3.13.2.18 MCUCNF17 Register (Offset = 200h) [Reset = X0000000h]

MCUCNF17 is shown in Figure 3-45 and described in Table 3-39.

Return to the Summary Table.

MCUCNF Capability: CLB

Figure 3-45. MCUCNF17 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		CLB6	CLB5	CLB4	CLB3	CLB2	CLB1
R-0h		R-0h	R-0h	R-Xh	R-Xh	R-Xh	R-Xh

Table 3-39. MCUCNF17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5	CLB6	R	0h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
4	CLB5	R	0h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
3	CLB4	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
2	CLB3	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
1	CLB2	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
0	CLB1	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn

3.13.2.19 MCUCNF18 Register (Offset = 204h) [Reset = 0000000Xh]

MCUCNF18 is shown in Figure 3-46 and described in Table 3-40.

Return to the Summary Table.

MCUCNF Capability: FSI

Figure 3-46. MCUCNF18 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				FSIRX_D	FSIRX_C	FSIRX_B	FSIRX_A
R-0h				R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				FSITX_D	FSITX_C	FSITX_B	FSITX_A
R-0-0h				R-Xh	R-Xh	R-Xh	R-Xh

Table 3-40. MCUCNF18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	FSIRX_D	R	0h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
18	FSIRX_C	R	0h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
17	FSIRX_B	R	0h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
16	FSIRX_A	R	0h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
15-4	RESERVED	R-0	0h	Reserved
3	FSITX_D	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
2	FSITX_C	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
1	FSITX_B	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
0	FSITX_A	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn

3.13.2.20 MCUCNF19 Register (Offset = 208h) [Reset = X0000000h]

MCUCNF19 is shown in Figure 3-47 and described in Table 3-41.

Return to the Summary Table.

MCUCNF Capability: LIN

Figure 3-47. MCUCNF19 Register

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

Table 3-41. MCUCNF19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3	RESERVED	R	Xh	Reserved
2	RESERVED	R	Xh	Reserved
1	LIN_B	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
0	LIN_A	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn

3.13.2.21 MCUCNF23 Register (Offset = 218h) [Reset = 0000000Xh]

MCUCNF23 is shown in Figure 3-48 and described in Table 3-42.

Return to the Summary Table.

MCUCNF Capability: EtherCAT

Figure 3-48. MCUCNF23 Register

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

Table 3-42. MCUCNF23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	ETHERCAT	R	Xh	ETHERCAT : 0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn

3.13.2.22 MCUCNF26 Register (Offset = 224h) [Reset = 00041041h]

MCUCNF26 is shown in Figure 3-49 and described in Table 3-43.

Return to the Summary Table.

Device Capability: HSM-Crypto Engines AES, SHA, PKA, TRNG

Figure 3-49. MCUCNF26 Register

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

Table 3-43. MCUCNF26 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-18	TRNG	R/W	1h	6'b1111111 : Module disabled 'others' : Module enabled Reset type: PORESETn
17-12	PKA	R/W	1h	6'b1111111 : Module disabled 'others' : Module enabled Reset type: PORESETn
11-6	SHA	R/W	1h	6'b1111111 : Module disabled 'others' : Module enabled Reset type: PORESETn
5-0	AES	R/W	1h	6'b1111111 : AES disabled 6'b101010 : AES enabled without counter measures 'others' : AES enabled with counter measures Reset type: PORESETn

3.13.2.23 MCUCNF31 Register (Offset = 238h) [Reset = 00001041h]

MCUCNF31 is shown in Figure 3-50 and described in Table 3-44.

Return to the Summary Table.

Device Capability: HSM-Crypto Engines SM2, SM3, SM4

Figure 3-50. MCUCNF31 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														SM4						SM3						SM2					
R-0h														R/W-1h						R/W-1h						R/W-1h					

Table 3-44. MCUCNF31 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17-12	SM4	R/W	1h	6'b1111111 : Module disabled 'others' : Module enabled Reset type: PORESETn
11-6	SM3	R/W	1h	6'b1111111 : Module disabled 'others' : Module enabled Reset type: PORESETn
5-0	SM2	R/W	1h	6'b1111111 : Module disabled 'others' : Module enabled Reset type: PORESETn

3.13.2.24 MCUCNF64 Register (Offset = 2BCh) [Reset = 0000001Xh]

MCUCNF64 is shown in Figure 3-51 and described in Table 3-45.

Return to the Summary Table.

MCUCNF Capability: MCUCNF level, Processing Block, RTDMA Customization

Figure 3-51. MCUCNF64 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED					RESERVED	RESERVED	RESERVED
R-0-0h					R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	CPU3	CPU2	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-0h	R/W-0h	R-Xh

Table 3-45. MCUCNF64 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	RESERVED	R/W	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	CPU3	R/W	1h	CPU Present 0: CPU3SS is not present 1: CPU3SS is present Reset type: PORESETn
3	CPU2	R/W	1h	CPU Present 0: CPU2SS is not present 1: CPU2SS is present Reset type: PORESETn
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R	Xh	Reserved

3.13.2.25 MCUCNF65 Register (Offset = 2C0h) [Reset = 0FFFFF3Xh]

MCUCNF65 is shown in Figure 3-52 and described in Table 3-46.

Return to the Summary Table.

MCUCNF Capability: On-chip SRAM Customization

Figure 3-52. MCUCNF65 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	CDA11	CDA10	CDA9	CDA8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
CDA7	CDA6	CDA5	CDA4	CDA3	CDA2	CDA1	CDA0
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
LDA7	LDA6	LDA5	LDA4	LDA3	LDA2	LDA1	LDA0
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	LPA1	LPA0	RESERVED	RESERVED	CPA1	CPA0
R-0-0h	R-0-0h	R/W-1h	R/W-1h	R-0-0h	R-0-0h	R/W-1h	R-Xh

Table 3-46. MCUCNF65 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	CDA11	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
26	CDA10	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
25	CDA9	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
24	CDA8	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
23	CDA7	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
22	CDA6	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
21	CDA5	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
20	CDA4	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn

Table 3-46. MCUCNF65 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	CDA3	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
18	CDA2	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
17	CDA1	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
16	CDA0	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
15	LDA7	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
14	LDA6	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
13	LDA5	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
12	LDA4	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
11	LDA3	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
10	LDA2	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
9	LDA1	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
8	LDA0	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
7	RESERVED	R-0	0h	Reserved
6	RESERVED	R-0	0h	Reserved
5	LPA1	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
4	LPA0	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
3	RESERVED	R-0	0h	Reserved
2	RESERVED	R-0	0h	Reserved
1	CPA1	R/W	1h	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn
0	CPA0	R	Xh	0: Feature not present on the device 1: Feature present on the device Reset type: PORESETn

3.13.2.26 MCUCNF74 Register (Offset = 2E4h) [Reset = 0000XXXXh]

MCUCNF74 is shown in Figure 3-53 and described in Table 3-47.

Return to the Summary Table.

MCUCNF Capability: FLC1.B0/B1

Figure 3-53. MCUCNF74 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
SECT255_240	SECT239_224	SECT223_208	SECT207_192	SECT191_176	SECT175_160	SECT159_144	SECT143_128
R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh
7	6	5	4	3	2	1	0
SECT127_112	SECT111_96	SECT95_80	SECT79_64	SECT63_48	SECT47_32	SECT31_16	SECT15_0
R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh

Table 3-47. MCUCNF74 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	SECT255_240	R	Xh	FLC1.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
14	SECT239_224	R	Xh	FLC1.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
13	SECT223_208	R	Xh	FLC1.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
12	SECT207_192	R	Xh	FLC1.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
11	SECT191_176	R	Xh	FLC1.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
10	SECT175_160	R	Xh	FLC1.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
9	SECT159_144	R	Xh	FLC1.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn

Table 3-47. MCUCNF74 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	SECT143_128	R	Xh	FLC1.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
7	SECT127_112	R	Xh	FLC1.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
6	SECT111_96	R	Xh	FLC1.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
5	SECT95_80	R	Xh	FLC1.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
4	SECT79_64	R	Xh	FLC1.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
3	SECT63_48	R	Xh	FLC1.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
2	SECT47_32	R	Xh	FLC1.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
1	SECT31_16	R	Xh	FLC1.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
0	SECT15_0	R	Xh	FLC1.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn

3.13.2.27 MCUCNF76 Register (Offset = 2ECh) [Reset = 0000XXXXh]

MCUCNF76 is shown in Figure 3-54 and described in Table 3-48.

Return to the Summary Table.

MCUCNF Capability: FLC1.B2/B3

Figure 3-54. MCUCNF76 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
SECT255_240	SECT239_224	SECT223_208	SECT207_192	SECT191_176	SECT175_160	SECT159_144	SECT143_128
R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh
7	6	5	4	3	2	1	0
SECT127_112	SECT111_96	SECT95_80	SECT79_64	SECT63_48	SECT47_32	SECT31_16	SECT15_0
R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh

Table 3-48. MCUCNF76 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	SECT255_240	R	Xh	FLC1.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
14	SECT239_224	R	Xh	FLC1.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
13	SECT223_208	R	Xh	FLC1.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
12	SECT207_192	R	Xh	FLC1.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
11	SECT191_176	R	Xh	FLC1.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
10	SECT175_160	R	Xh	FLC1.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
9	SECT159_144	R	Xh	FLC1.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn

Table 3-48. MCUCNF76 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	SECT143_128	R	Xh	FLC1.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
7	SECT127_112	R	Xh	FLC1.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
6	SECT111_96	R	Xh	FLC1.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
5	SECT95_80	R	Xh	FLC1.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
4	SECT79_64	R	Xh	FLC1.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
3	SECT63_48	R	Xh	FLC1.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
2	SECT47_32	R	Xh	FLC1.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
1	SECT31_16	R	Xh	FLC1.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
0	SECT15_0	R	Xh	FLC1.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn

3.13.2.28 MCUCNF78 Register (Offset = 2F4h) [Reset = 0000XXXXh]

MCUCNF78 is shown in Figure 3-55 and described in Table 3-49.

Return to the Summary Table.

MCUCNF Capability: FLC1.B4 256KB Data Flash

Figure 3-55. MCUCNF78 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh
7	6	5	4	3	2	1	0
SECT127_112	SECT111_96	SECT95_80	SECT79_64	SECT63_48	SECT47_32	SECT31_16	SECT15_0
R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh

Table 3-49. MCUCNF78 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	RESERVED	R	Xh	Reserved
14	RESERVED	R	Xh	Reserved
13	RESERVED	R	Xh	Reserved
12	RESERVED	R	Xh	Reserved
11	RESERVED	R	Xh	Reserved
10	RESERVED	R	Xh	Reserved
9	RESERVED	R	Xh	Reserved
8	RESERVED	R	Xh	Reserved
7	SECT127_112	R	Xh	FLC1.B4: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
6	SECT111_96	R	Xh	FLC1.B4: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
5	SECT95_80	R	Xh	FLC1.B4: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
4	SECT79_64	R	Xh	FLC1.B4: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
3	SECT63_48	R	Xh	FLC1.B4: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn

Table 3-49. MCUCNF78 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	SECT47_32	R	Xh	FLC1.B4: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
1	SECT31_16	R	Xh	FLC1.B4: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
0	SECT15_0	R	Xh	FLC1.B4: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn

3.13.2.29 MCUCNF79 Register (Offset = 2F8h) [Reset = 0000XXXXh]

MCUCNF79 is shown in Figure 3-56 and described in Table 3-50.

Return to the Summary Table.

MCUCNF Capability: FLC2.B0/B1

Figure 3-56. MCUCNF79 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
SECT255_240	SECT239_224	SECT223_208	SECT207_192	SECT191_176	SECT175_160	SECT159_144	SECT143_128
R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh
7	6	5	4	3	2	1	0
SECT127_112	SECT111_96	SECT95_80	SECT79_64	SECT63_48	SECT47_32	SECT31_16	SECT15_0
R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh

Table 3-50. MCUCNF79 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	SECT255_240	R	Xh	FLC2.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
14	SECT239_224	R	Xh	FLC2.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
13	SECT223_208	R	Xh	FLC2.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
12	SECT207_192	R	Xh	FLC2.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
11	SECT191_176	R	Xh	FLC2.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
10	SECT175_160	R	Xh	FLC2.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
9	SECT159_144	R	Xh	FLC2.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn

Table 3-50. MCUCNF79 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	SECT143_128	R	Xh	FLC2.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
7	SECT127_112	R	Xh	FLC2.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
6	SECT111_96	R	Xh	FLC2.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
5	SECT95_80	R	Xh	FLC2.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
4	SECT79_64	R	Xh	FLC2.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
3	SECT63_48	R	Xh	FLC2.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
2	SECT47_32	R	Xh	FLC2.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
1	SECT31_16	R	Xh	FLC2.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
0	SECT15_0	R	Xh	FLC2.B0/B1: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn

3.13.2.30 MCUCNF81 Register (Offset = 300h) [Reset = 0000XXXXh]

MCUCNF81 is shown in Figure 3-57 and described in Table 3-51.

Return to the Summary Table.

MCUCNF Capability: FLC2.B2/B3

Figure 3-57. MCUCNF81 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
SECT255_240	SECT239_224	SECT223_208	SECT207_192	SECT191_176	SECT175_160	SECT159_144	SECT143_128
R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh
7	6	5	4	3	2	1	0
SECT127_112	SECT111_96	SECT95_80	SECT79_64	SECT63_48	SECT47_32	SECT31_16	SECT15_0
R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh	R-Xh

Table 3-51. MCUCNF81 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	SECT255_240	R	Xh	FLC2.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
14	SECT239_224	R	Xh	FLC2.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
13	SECT223_208	R	Xh	FLC2.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
12	SECT207_192	R	Xh	FLC2.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
11	SECT191_176	R	Xh	FLC2.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
10	SECT175_160	R	Xh	FLC2.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
9	SECT159_144	R	Xh	FLC2.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn

Table 3-51. MCUCNF81 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	SECT143_128	R	Xh	FLC2.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
7	SECT127_112	R	Xh	FLC2.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
6	SECT111_96	R	Xh	FLC2.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
5	SECT95_80	R	Xh	FLC2.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
4	SECT79_64	R	Xh	FLC2.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
3	SECT63_48	R	Xh	FLC2.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
2	SECT47_32	R	Xh	FLC2.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
1	SECT31_16	R	Xh	FLC2.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
0	SECT15_0	R	Xh	FLC2.B2/B3: 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn

3.13.2.31 MCUCNFLOCK1 Register (Offset = 33Ch) [Reset = 00000000h]

MCUCNFLOCK1 is shown in Figure 3-58 and described in Table 3-52.

Return to the Summary Table.

Lock bit for MCUCNFx registers
The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed
Note:
Any SOnce bit in this register, once set can only be cleared through a CPU1.SYSRSn. Write of 0 to any bit of this register has no effect

Figure 3-58. MCUCNFLOCK1 Register

31	30	29	28	27	26	25	24
MCUCNF31	RESERVED				MCUCNF26	RESERVED	
R/WSonce-0h	R-0-0h				R/WSonce-0h	R-0-0h	
23	22	21	20	19	18	17	16
MCUCNF23	RESERVED			MCUCNF19	MCUCNF18	MCUCNF17	RESERVED
R/WSonce-0h	R-0-0h			R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R-0-0h
15	14	13	12	11	10	9	8
RESERVED					MCUCNF10	RESERVED	
R-0-0h					R/WSonce-0h	R-0-0h	
7	6	5	4	3	2	1	0
RESERVED			MCUCNF4	RESERVED	MCUCNF2	MCUCNF1	RESERVED
R-0-0h			R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R-0-0h

Table 3-52. MCUCNFLOCK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MCUCNF31	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
30-27	RESERVED	R-0	0h	Reserved
26	MCUCNF26	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
25-24	RESERVED	R-0	0h	Reserved
23	MCUCNF23	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
22-20	RESERVED	R-0	0h	Reserved
19	MCUCNF19	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
18	MCUCNF18	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn

Table 3-52. MCUCNFLOCK1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	MCUCNF17	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
16-11	RESERVED	R-0	0h	Reserved
10	MCUCNF10	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
9-5	RESERVED	R-0	0h	Reserved
4	MCUCNF4	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
3	RESERVED	R/WSonce	0h	Reserved
2	MCUCNF2	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
1	MCUCNF1	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
0	RESERVED	R-0	0h	Reserved

3.13.2.32 MCUCNFLOCK2 Register (Offset = 340h) [Reset = 00000000h]

MCUCNFLOCK2 is shown in Figure 3-59 and described in Table 3-53.

Return to the Summary Table.

Lock bit for MCUCNFx registers
The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed
Note:
Any SOnce bit in this register, once set can only be cleared through a CPU1.SYSRSn. Write of 0 to any bit of this register has no effect

Figure 3-59. MCUCNFLOCK2 Register

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

Table 3-53. MCUCNFLOCK2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/WSonce	0h	Reserved

3.13.2.33 MCUCNFLOCK3 Register (Offset = 344h) [Reset = 00000000h]

MCUCNFLOCK3 is shown in Figure 3-60 and described in Table 3-54.

Return to the Summary Table.

Lock bit for MCUCNFx registers
The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed
Note:
Any SOnce bit in this register, once set can only be cleared through a CPU1.SYSRSn. Write of 0 to any bit of this register has no effect

Figure 3-60. MCUCNFLOCK3 Register

31	30	29	28	27	26	25	24
RESERVED	MCUCNF95	RESERVED					
R/WSonce-0h	R/WSonce-0h	R-0-0h					
23	22	21	20	19	18	17	16
RESERVED						MCUCNF81	RESERVED
R-0-0h						R/WSonce-0h	R-0-0h
15	14	13	12	11	10	9	8
MCUCNF79	MCUCNF78	RESERVED	MCUCNF76	RESERVED	MCUCNF74	RESERVED	RESERVED
R/WSonce-0h	R/WSonce-0h	R-0-0h	R/WSonce-0h	R-0-0h	R/WSonce-0h	R-0-0h	R-0-0h
7	6	5	4	3	2	1	0
RESERVED						MCUCNF65	MCUCNF64
R/WSonce-0h						R/WSonce-0h	R/WSonce-0h

Table 3-54. MCUCNFLOCK3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/WSonce	0h	Reserved
30	MCUCNF95	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
29-18	RESERVED	R-0	0h	Reserved
17	MCUCNF81	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
16	RESERVED	R-0	0h	Reserved
15	MCUCNF79	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
14	MCUCNF78	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
13	RESERVED	R-0	0h	Reserved
12	MCUCNF76	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
11	RESERVED	R-0	0h	Reserved

Table 3-54. MCUCNFLOCK3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	MCUCNF74	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
9	RESERVED	R-0	0h	Reserved
8	RESERVED	R-0	0h	Reserved
7-2	RESERVED	R/WSonce	0h	Reserved
1	MCUCNF65	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
0	MCUCNF64	R/WSonce	0h	Lock bit for MCUCNF register: 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn

3.13.2.34 LSEN Register (Offset = 348h) [Reset = 00000001h]

LSEN is shown in Figure 3-61 and described in Table 3-55.

Return to the Summary Table.

Lockstep enable configuration

Figure 3-61. LSEN Register

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

Table 3-55. LSEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Rserved	R-0	0h	Reserved Reset type: PORESETn
0	Enable	R/W	1h	0: Lockstep is disabled 1: Lockstep is enabled Note: User is expected to lock and commit the specific configuration as inadvertent clearing of the bit will cause lockstep to be disabled. Reset type: PORESETn

3.13.2.35 EPWMXLINKCFG Register (Offset = 37Ch) [Reset = 00000000h]

EPWMXLINKCFG is shown in Figure 3-62 and described in Table 3-56.

Return to the Summary Table.

Configure which EPWM module instaces are linked in the XLINK scheme

Figure 3-62. EPWMXLINKCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						EPWM18	EPWM17
R-0-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
EPWM16	EPWM15	EPWM14	EPWM13	EPWM12	EPWM11	EPWM10	EPWM9
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
EPWM8	EPWM7	EPWM6	EPWM5	EPWM4	EPWM3	EPWM2	EPWM1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-56. EPWMXLINKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	Reserved
17	EPWM18	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn
16	EPWM17	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn
15	EPWM16	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn
14	EPWM15	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn
13	EPWM14	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn
12	EPWM13	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn
11	EPWM12	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn

Table 3-56. EPWMXLINKCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	EPWM11	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn
9	EPWM10	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn
8	EPWM9	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn
7	EPWM8	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn
6	EPWM7	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn
5	EPWM6	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn
4	EPWM5	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn
3	EPWM4	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn
2	EPWM3	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn
1	EPWM2	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn
0	EPWM1	R/W	0h	Selects Peripheral instance mirrored to the XLINK Region 0: Disabled - Module instance is not mirrored to XLINK 1: Enabled - Module instance is mirrored to XLINK Reset type: CPU1.SYSRSn

3.13.2.36 SICCONFIG Register (Offset = 384h) [Reset = FFFF0000h]

SICCONFIG is shown in Figure 3-63 and described in Table 3-57.

Return to the Summary Table.

Safety Interconnect(SIC) Configuration - Enable and READY TIMEOUT value

Figure 3-63. SICCONFIG Register

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

Table 3-57. SICCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TIMEOUT	R/W	FFFFh	Safety Interconnect (SIC) READY TIMEOUT value, in terms of number of clock cycles. This is a 16-bit value common across multiple CPU, DMA or any other initiator on the inyterconnect. This value controls the input to the C29 CPU/SIC to detect bus-hang condition. Due to a fault, if ready is pulled low by the endpoint longer than number of clock cycles programmed in this register, then timeout logic inside SIC will generate abort access to endpoint and aborts the ongoing access on bus, communicates as error on the error interface bus. Reset type: XRSn
15-1	RESERVED	R-0	0h	Reserved
0	Enable	R/W	0h	Safety Interconnect (SIC) Enable Reset type: XRSn

3.13.2.37 RSTSTAT Register (Offset = 3B0h) [Reset = 00000000h]

RSTSTAT is shown in Figure 3-64 and described in Table 3-58.

Return to the Summary Table.

Reset Status register for secondary CPUs

Figure 3-64. RSTSTAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				RESERVED		RESERVED	RESERVED
R-0-0h				R/W1S-0h		R/W1S-0h	R/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CPU3	CPU2
R/W1S-0h	R/W1S-0h	R/W1S-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 3-58. RSTSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	RESERVED	R/W1S	0h	Reserved
9	RESERVED	R/W1S	0h	Reserved
8	RESERVED	R/W1S	0h	Reserved
7	RESERVED	R/W1S	0h	Reserved
6	RESERVED	R/W1S	0h	Reserved
5	RESERVED	R/W1S	0h	Reserved
4	RESERVED	R	0h	Reserved
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	CPU3	R	0h	Reset status of CPU3 to CPU1 0: CPU3 core is in reset 1: CPU3 core is out of reset Reset type: CPU1.SYSRSn
0	CPU2	R	0h	Reset status of CPU2 to CPU1 0: CPU2 core is in reset 1: CPU2 core is out of reset Reset type: CPU1.SYSRSn

3.13.2.38 LPMSTAT Register (Offset = 3B4h) [Reset = 00000000h]

LPMSTAT is shown in Figure 3-65 and described in Table 3-59.

Return to the Summary Table.

LPM Status Register for secondary CPUs

Figure 3-65. LPMSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						RESERVED		RESERVED		RESERVED		CPU3		CPU2	
R-0-0h						R-0h		R-0h		R-0h		R-0h		R-0h	

Table 3-59. LPMSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R-0	0h	Reserved
9-8	RESERVED	R	0h	Reserved
7-6	RESERVED	R	0h	Reserved
5-4	RESERVED	R	0h	Reserved
3-2	CPU3	R	0h	These bits indicate the power mode CPU3 00: CPU is in ACTIVE mode 01: CPU is in IDLE mode 10: CPU is in STANDBY mode 11: Reserved Reset type: CPU1.SYSRSn
1-0	CPU2	R	0h	These bits indicate the power mode CPU2 00: CPU is in ACTIVE mode 01: CPU is in IDLE mode 10: CPU is in STANDBY mode 11: Reserved Reset type: CPU1.SYSRSn

3.13.2.39 TAP_STATUS Register (Offset = 3CCh) [Reset = 00000000h]

TAP_STATUS is shown in Figure 3-66 and described in Table 3-60.

Return to the Summary Table.

Status of JTAG State machine & Debugger Connect

Figure 3-66. TAP_STATUS Register

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

Table 3-60. TAP_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DCON	R	0h	DebugConnect indication from IcePick. Reset type: PORESETn
30-16	RESERVED	R-0	0h	Reserved
15-0	TAP_STATE	R	0h	TAP State Vector. With bits representing, Connect coresponding POTAP* output to the 0:TLR, 1:IDLE, 2:SELECTDR, 3:CAPDR, 4:SHIFTDR, 5:EXIT1DR, 6:PAUSEDR, 7:EXIT2DR, 8:UPDTDR, 9:SLECTIR, 10:CAPIR, 11:SHIFTIR, 12:EXIT1IR, 13:PAUSEIR, 14:EXIT2IR, 15:UPDTIR, Reset type: PORESETn

3.13.2.40 TAP_CONTROL Register (Offset = 3D0h) [Reset = 00000000h]

TAP_CONTROL is shown in Figure 3-67 and described in Table 3-61.

Return to the Summary Table.

Disable TAP control

Figure 3-67. TAP_CONTROL Register

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

Table 3-61. TAP_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write to this register succeeds only if this field is written with a value of 0xa5a5 Note: [1] Due to this KEY, only 32-bit writes will succeed (provided the KEY matches). 16-bit writes to the upper or lower half of this register will be ignored Reset type: PORESETn
15-1	RESERVED	R-0	0h	Reserved
0	BSCAN_DIS	R/W	0h	Disables BSCAN TAP control : 0: BSCAN TAP control enabled 1: BSCAN TAP control disabled Reset type: PORESETn

3.13.2.41 DEVLIFECYCLE Register (Offset = 3D4h) [Reset = 00000000h]

DEVLIFECYCLE is shown in Figure 3-68 and described in Table 3-62.

Return to the Summary Table.

Reflect the state of the Device Life Cycle signals reported from the HSM

Figure 3-68. DEVLIFECYCLE Register

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

Table 3-62. DEVLIFECYCLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	Reserved
17	OVRNOFLASH	R	0h	TI Override mode with no access to Flash Reset type: PORESETn
16	OVRFLASH	R	0h	TI Override mode with access to Flash Reset type: PORESETn
15-12	RESERVED	R-0	0h	Reserved
11-8	HSSUBTYPE	R	0h	These bits reflect the state of the signals from the HSM DEVICE_HS_SUBTYPE Field values: 4'b1010: FS (Field Securable) 4'b0011: KP (Keys Provisioned) 4'b1111: FA (Failure Analysis) Other: SE (Security Enforced) Reset type: PORESETn
7-4	RESERVED	R-0	0h	Reserved
3-0	DEVTYPE	R	0h	These bits reflect the state of the signals from the HSM DEVICE_TYPE Field values: 4'b0101: TEST 4'b1001: EMU (EMULATOR) 4'b1010: HS (HIGH_SECURITY) 4'b0011: GP (GENERAL_PURPOSE) Other: BAD Reset type: PORESETn

3.13.2.42 SDFMTYPE Register (Offset = 47Ch) [Reset = 00000000h]

SDFMTYPE is shown in Figure 3-69 and described in Table 3-63.

Return to the Summary Table.

Based on the configuration enables disables features associated with the SDFM type.

Figure 3-69. SDFMTYPE Register

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

Table 3-63. SDFMTYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	LOCK	R/WSonce	0h	1: Write to this register is not allowed. 0: Write to this register is allowed. Reset type: CPU1.SYSRSn
14-2	RESERVED	R-0	0h	Reserved
1-0	TYPE	R/W	0h	'00,10,11' : 1. Data Ready conditions combined with the fault conditions on the SDFM interrupt line. 2. Data ready interrupts from individual filters are not generated. '01' : 1. Data Ready conditions do not generate the SDFMINT. 2. Each filter generates a separate data ready interrupts. Reset type: CPU1.SYSRSn

3.13.2.43 SYNCSELECT Register (Offset = 4ACh) [Reset = 00000000h]

SYNCSELECT is shown in Figure 3-70 and described in Table 3-64.

Return to the Summary Table.

Sync Input and Output Select Register

Figure 3-70. SYNCSELECT Register

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

Table 3-64. SYNCSELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved

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

Bit	Field	Type	Reset	Description
28-24	SYNCOUT	R/W	0h	Select Syncout Source: 00000: EPWM1SYNCOUT selected to drive the SYNCOUT pin. 00001: EPWM2SYNCOUT selected to drive the SYNCOUT pin. 00010: EPWM3SYNCOUT selected to drive the SYNCOUT pin. 00011: EPWM4SYNCOUT selected to drive the SYNCOUT pin. 00100: EPWM5SYNCOUT selected to drive the SYNCOUT pin. 00101: EPWM6SYNCOUT selected to drive the SYNCOUT pin. 00110: EPWM7SYNCOUT selected to drive the SYNCOUT pin. 00111: EPWM8SYNCOUT selected to drive the SYNCOUT pin. 01000: EPWM9SYNCOUT selected to drive the SYNCOUT pin. 01001: EPWM10SYNCOUT selected to drive the SYNCOUT pin. 01010: EPWM11SYNCOUT selected to drive the SYNCOUT pin. 01011: EPWM12SYNCOUT selected to drive the SYNCOUT pin. 01100: EPWM13SYNCOUT selected to drive the SYNCOUT pin. 01101: EPWM14SYNCOUT selected to drive the SYNCOUT pin. 01110: EPWM15SYNCOUT selected to drive the SYNCOUT pin. 01111: EPWM16SYNCOUT selected to drive the SYNCOUT pin. 10000: EPWM17SYNCOUT selected to drive the SYNCOUT pin. 10001: EPWM18SYNCOUT selected to drive the SYNCOUT pin. 10010: Reserved 10011: Reserved 10100: Reserved 10101: Reserved 10110: Reserved 10111: Reserved 11000: ECAP1SYNCOUT selected to drive the SYNCOUT pin. 11001: ECAP2SYNCOUT selected to drive the SYNCOUT pin. 11010: ECAP3SYNCOUT selected to drive the SYNCOUT pin. 11011: ECAP4SYNCOUT selected to drive the SYNCOUT pin. 11100: ECAP5SYNCOUT selected to drive the SYNCOUT pin. 11101: ECAP6SYNCOUT selected to drive the SYNCOUT pin. 11110: Reserved. 11111: Reserved Notes: [1] Reserved position defaults to 00 selection Reset type: CPU1.SYSRSn
23-18	RESERVED	R-0	0h	Reserved
17-15	RESERVED	R/W	0h	Reserved
14-12	RESERVED	R/W	0h	Reserved
11-9	RESERVED	R/W	0h	Reserved
8-6	RESERVED	R/W	0h	Reserved
5-3	RESERVED	R/W	0h	Reserved
2-0	RESERVED	R/W	0h	Reserved

3.13.2.44 ADCSOCOUTSELECT Register (Offset = 4B0h) [Reset = 00000000h]

ADCSOCOUTSELECT is shown in Figure 3-71 and described in Table 3-65.

Return to the Summary Table.

External ADCSOC Select Register (PWM1-16)

Figure 3-71. ADCSOCOUTSELECT Register

31	30	29	28	27	26	25	24
PWM16SOBAEN	PWM15SOBAEN	PWM14SOBAEN	PWM13SOCBEN	PWM12SOBAEN	PWM11SOBAEN	PWM10SOBAEN	PWM9SOCBEN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
PWM8SOCBEN	PWM7SOCBEN	PWM6SOCBEN	PWM5SOCBEN	PWM4SOCBEN	PWM3SOCBEN	PWM2SOCBEN	PWM1SOCBEN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
PWM16SOCAEN	PWM15SOCAEN	PWM14SOCAEN	PWM13SOCAEN	PWM12SOCAEN	PWM11SOCAEN	PWM10SOCAEN	PWM9SOCAEN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
PWM8SOCAEN	PWM7SOCAEN	PWM6SOCAEN	PWM5SOCAEN	PWM4SOCAEN	PWM3SOCAEN	PWM2SOCAEN	PWM1SOCAEN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-65. ADCSOCOUTSELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PWM16SOBAEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
30	PWM15SOBAEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
29	PWM14SOBAEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
28	PWM13SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
27	PWM12SOBAEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
26	PWM11SOBAEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
25	PWM10SOBAEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn

Table 3-65. ADCSOCOUTSELECT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	PWM9SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
23	PWM8SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
22	PWM7SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
21	PWM6SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
20	PWM5SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
19	PWM4SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
18	PWM3SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
17	PWM2SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
16	PWM1SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
15	PWM16SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn
14	PWM15SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn
13	PWM14SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn
12	PWM13SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn
11	PWM12SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn

Table 3-65. ADCSOCOUTSELECT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	PWM11SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn
9	PWM10SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn
8	PWM9SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn
7	PWM8SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn
6	PWM7SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn
5	PWM6SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn
4	PWM5SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn
3	PWM4SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn
2	PWM3SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn
1	PWM2SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn
0	PWM1SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn

3.13.2.45 ADCSOCOUTSELECT1 Register (Offset = 4B4h) [Reset = 00000000h]

ADCSOCOUTSELECT1 is shown in Figure 3-72 and described in Table 3-66.

Return to the Summary Table.

External ADCSOC Select Register (PWM17-32)

Figure 3-72. ADCSOCOUTSELECT1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						PWM18SOCBEN	PWM17SOCBEN
R-0-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						PWM18SOCAEN	PWM17SOCAEN
R-0-0h						R/W-0h	R/W-0h

Table 3-66. ADCSOCOUTSELECT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	Reserved
17	PWM18SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
16	PWM17SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective EPWM SOCB output is not selected 1: Respective EPWM SOCB output is selected Reset type: CPU1.SYSRSn
15-2	RESERVED	R-0	0h	Reserved
1	PWM18SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn
0	PWM17SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective EPWM SOCA output is not selected 1: Respective EPWM SOCA output is selected Reset type: CPU1.SYSRSn

3.13.2.46 SYNCSOCLOCK Register (Offset = 4B8h) [Reset = 00000000h]

SYNCSOCLOCK is shown in Figure 3-73 and described in Table 3-67.

Return to the Summary Table.

SYNCSEL and ADCSOC Select Lock register

Figure 3-73. SYNCSOCLOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					ADCSOCOUTSELECT1	ADCSOCOUTSELECT	SYNCSELECT
R-0-0h					R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 3-67. SYNCSOCLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	ADCSOCOUTSELECT1	R/WSonce	0h	ADCSOCOUTSELECT1 Register Lock bit: 0: Respective register is not locked 1: Respective register is locked. Notes: [1] Any bit in this register, once set can only be creaed through a SYSRSn. Write of 0 to any bit of this regtister has no effect [2] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed Reset type: CPU1.SYSRSn
1	ADCSOCOUTSELECT	R/WSonce	0h	ADCSOCOUTSELECT Register Lock bit: 0: Respective register is not locked 1: Respective register is locked. Notes: [1] Any bit in this register, once set can only be creaed through a SYSRSn. Write of 0 to any bit of this regtister has no effect [2] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed Reset type: CPU1.SYSRSn
0	SYNCSELECT	R/WSonce	0h	SYNCSELECT Register Lock bit: 0: Respective register is not locked 1: Respective register is locked. Notes: [1] Any bit in this register, once set can only be creaed through a SYSRSn. Write of 0 to any bit of this regtister has no effect [2] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed Reset type: CPU1.SYSRSn

3.13.2.47 HSMTOCPU_STS1 Register (Offset = 4DCh) [Reset = 00000000h]

HSMTOCPU_STS1 is shown in Figure 3-74 and described in Table 3-68.

Return to the Summary Table.

Communicate from HSM to CPU control signal

Figure 3-74. HSMTOCPU_STS1 Register

31	30	29	28	27	26	25	24
SOC_GENR_2							
R-0h							
23	22	21	20	19	18	17	16
SOC_GENR_1							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				MPOST	LPOST	FLC2	FLC1
R-0-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
SRAM_BANK7	SRAM_BANK6	SRAM_BANK5	SRAM_BANK4	SRAM_BANK3	SRAM_BANK2	SRAM_BANK1	SRAM_BANK0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 3-68. HSMTOCPU_STS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOC_GENR_2	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
23-16	SOC_GENR_1	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
15-12	RESERVED	R-0	0h	Reserved
11	MPOST	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
10	LPOST	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
9	FLC2	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
8	FLC1	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
7	SRAM_BANK7	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
6	SRAM_BANK6	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
5	SRAM_BANK5	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn

Table 3-68. HSMTOCPU_STS1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	SRAM_BANK4	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
3	SRAM_BANK3	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
2	SRAM_BANK2	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
1	SRAM_BANK1	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
0	SRAM_BANK0	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn

3.13.2.48 HSMTOCPU_STS2 Register (Offset = 4E0h) [Reset = 00000000h]

HSMTOCPU_STS2 is shown in Figure 3-75 and described in Table 3-69.

Return to the Summary Table.

Communicate from HSM to CPU control signal

Figure 3-75. HSMTOCPU_STS2 Register

31	30	29	28	27	26	25	24
RESERVED							FLC2_BANK4
R-0-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED			FLC1_BANK4	FLC2_BANK3	RESERVED		
R-0-0h			R-0h	R-0h	R-0-0h		
15	14	13	12	11	10	9	8
FLC1_BANK3	FLC2_BANK2	RESERVED			FLC1_BANK2	FLC2_BANK1	RESERVED
R-0h	R-0h	R-0-0h			R-0h	R-0h	R-0-0h
7	6	5	4	3	2	1	0
RESERVED		FLC1_BANK1	FLC2_BANK0	RESERVED			FLC1_BANK0
R-0-0h		R-0h	R-0h	R-0-0h			R-0h

Table 3-69. HSMTOCPU_STS2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R-0	0h	Reserved
24	FLC2_BANK4	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
23-21	RESERVED	R-0	0h	Reserved
20	FLC1_BANK4	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
19	FLC2_BANK3	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
18-16	RESERVED	R-0	0h	Reserved
15	FLC1_BANK3	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
14	FLC2_BANK2	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
13-11	RESERVED	R-0	0h	Reserved
10	FLC1_BANK2	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
9	FLC2_BANK1	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
8-6	RESERVED	R-0	0h	Reserved
5	FLC1_BANK1	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn

Table 3-69. HSMTOCPU_STS2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	FLC2_BANK0	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn
3-1	RESERVED	R-0	0h	Reserved
0	FLC1_BANK0	R	0h	Read-Only SIGNAL BIT for direct communication from HSM to CPU, to be used for handshake during Boot or Runtime Reset type: CPU1.SYSRSn

3.13.2.49 HSM_SECURE_BOOT_INFO_REG0 Register (Offset = 4E4h) [Reset = 00000000h]

HSM_SECURE_BOOT_INFO_REG0 is shown in Figure 3-76 and described in Table 3-70.

Return to the Summary Table.

Communicate from HSM to CPU1 during secure Boot

Figure 3-76. HSM_SECURE_BOOT_INFO_REG0 Register

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

Table 3-70. HSM_SECURE_BOOT_INFO_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STATUS	R	0h	These set of SW registers (HSM_SECURE_BOOT_INFO_REG0-7) hold the status information of the secure boot process for communication with the Host application. The definitions of the register fields (part of HSM_SOC_CTRL address space) are dependent on the implementation of the secure boot flow. These registers captures the output from the HSM module to the SoC. Reset type: PORESETn

3.13.2.50 HSM_SECURE_BOOT_INFO_REG1 Register (Offset = 4E8h) [Reset = 00000000h]

HSM_SECURE_BOOT_INFO_REG1 is shown in Figure 3-77 and described in Table 3-71.

Return to the Summary Table.

Communicate from HSM to CPU1 during secure Boot

Figure 3-77. HSM_SECURE_BOOT_INFO_REG1 Register

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

Table 3-71. HSM_SECURE_BOOT_INFO_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STATUS	R	0h	These set of SW registers (HSM_SECURE_BOOT_INFO_REG0-7) hold the status information of the secure boot process for communication with the Host application. The definitions of the register fields (part of HSM_SOC_CTRL address space) are dependent on the implementation of the secure boot flow. These registers captures the output from the HSM module to the SoC. Reset type: PORESETn

3.13.2.51 HSM_SECURE_BOOT_INFO_REG2 Register (Offset = 4ECh) [Reset = 00000000h]

HSM_SECURE_BOOT_INFO_REG2 is shown in Figure 3-78 and described in Table 3-72.

Return to the Summary Table.

Communicate from HSM to CPU1 during secure Boot

Figure 3-78. HSM_SECURE_BOOT_INFO_REG2 Register

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

Table 3-72. HSM_SECURE_BOOT_INFO_REG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STATUS	R	0h	These set of SW registers (HSM_SECURE_BOOT_INFO_REG0-7) hold the status information of the secure boot process for communication with the Host application. The definitions of the register fields (part of HSM_SOC_CTRL address space) are dependent on the implementation of the secure boot flow. These registers captures the output from the HSM module to the SoC. Reset type: PORESETn

3.13.2.52 HSM_SECURE_BOOT_INFO_REG3 Register (Offset = 4F0h) [Reset = 00000000h]

HSM_SECURE_BOOT_INFO_REG3 is shown in Figure 3-79 and described in Table 3-73.

Return to the Summary Table.

Communicate from HSM to CPU1 during secure Boot

Figure 3-79. HSM_SECURE_BOOT_INFO_REG3 Register

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

Table 3-73. HSM_SECURE_BOOT_INFO_REG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STATUS	R	0h	These set of SW registers (HSM_SECURE_BOOT_INFO_REG0-7) hold the status information of the secure boot process for communication with the Host application. The definitions of the register fields (part of HSM_SOC_CTRL address space) are dependent on the implementation of the secure boot flow. These registers captures the output from the HSM module to the SoC. Reset type: PORESETn

3.13.2.53 HSM_SECURE_BOOT_INFO_REG4 Register (Offset = 4F4h) [Reset = 00000000h]

HSM_SECURE_BOOT_INFO_REG4 is shown in Figure 3-80 and described in Table 3-74.

Return to the Summary Table.

Communicate from HSM to CPU1 during secure Boot

Figure 3-80. HSM_SECURE_BOOT_INFO_REG4 Register

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

Table 3-74. HSM_SECURE_BOOT_INFO_REG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STATUS	R	0h	These set of SW registers (HSM_SECURE_BOOT_INFO_REG0-7) hold the status information of the secure boot process for communication with the Host application. The definitions of the register fields (part of HSM_SOC_CTRL address space) are dependent on the implementation of the secure boot flow. These registers captures the output from the HSM module to the SoC. Reset type: PORESETn

3.13.2.54 HSM_SECURE_BOOT_INFO_REG5 Register (Offset = 4F8h) [Reset = 00000000h]

HSM_SECURE_BOOT_INFO_REG5 is shown in Figure 3-81 and described in Table 3-75.

Return to the Summary Table.

Communicate from HSM to CPU1 during secure Boot

Figure 3-81. HSM_SECURE_BOOT_INFO_REG5 Register

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

Table 3-75. HSM_SECURE_BOOT_INFO_REG5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STATUS	R	0h	These set of SW registers (HSM_SECURE_BOOT_INFO_REG0-7) hold the status information of the secure boot process for communication with the Host application. The definitions of the register fields (part of HSM_SOC_CTRL address space) are dependent on the implementation of the secure boot flow. These registers captures the output from the HSM module to the SoC. Reset type: PORESETn

3.13.2.55 HSM_SECURE_BOOT_INFO_REG6 Register (Offset = 4FCh) [Reset = 00000000h]

HSM_SECURE_BOOT_INFO_REG6 is shown in Figure 3-82 and described in Table 3-76.

Return to the Summary Table.

Communicate from HSM to CPU1 during secure Boot

Figure 3-82. HSM_SECURE_BOOT_INFO_REG6 Register

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

Table 3-76. HSM_SECURE_BOOT_INFO_REG6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STATUS	R	0h	These set of SW registers (HSM_SECURE_BOOT_INFO_REG0-7) hold the status information of the secure boot process for communication with the Host application. The definitions of the register fields (part of HSM_SOC_CTRL address space) are dependent on the implementation of the secure boot flow. These registers captures the output from the HSM module to the SoC. Reset type: PORESETn

3.13.2.56 HSM_SECURE_BOOT_INFO_REG7 Register (Offset = 500h) [Reset = 00000000h]

HSM_SECURE_BOOT_INFO_REG7 is shown in Figure 3-83 and described in Table 3-77.

Return to the Summary Table.

Communicate from HSM to CPU1 during secure Boot

Figure 3-83. HSM_SECURE_BOOT_INFO_REG7 Register

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

Table 3-77. HSM_SECURE_BOOT_INFO_REG7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STATUS	R	0h	These set of SW registers (HSM_SECURE_BOOT_INFO_REG0-7) hold the status information of the secure boot process for communication with the Host application. The definitions of the register fields (part of HSM_SOC_CTRL address space) are dependent on the implementation of the secure boot flow. These registers captures the output from the HSM module to the SoC. Reset type: PORESETn

3.13.2.57 SOC_SECURE_BOOT_INFO_REG0 Register (Offset = 504h) [Reset = 00000000h]

SOC_SECURE_BOOT_INFO_REG0 is shown in Figure 3-84 and described in Table 3-78.

Return to the Summary Table.

Communicate from CPU1 to HSM during secure Boot

Figure 3-84. SOC_SECURE_BOOT_INFO_REG0 Register

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

Table 3-78. SOC_SECURE_BOOT_INFO_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STATUS	R/W	0h	These set of SW registers (SOC_SECURE_BOOT_INFO_REG0-7) hold the status information of the secure boot process for communication with the Host application. The definitions of the register fields (part of HSM_SOC_CTRL address space) are dependent on the implementation of the secure boot flow. These registers captures the output from the SoC system control module to the HSM. Reset type: PORESETn

3.13.2.58 SOC_SECURE_BOOT_INFO_REG1 Register (Offset = 508h) [Reset = 00000000h]

SOC_SECURE_BOOT_INFO_REG1 is shown in Figure 3-85 and described in Table 3-79.

Return to the Summary Table.

Communicate from CPU1 to HSM during secure Boot

Figure 3-85. SOC_SECURE_BOOT_INFO_REG1 Register

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

Table 3-79. SOC_SECURE_BOOT_INFO_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STATUS	R/W	0h	These set of SW registers (SOC_SECURE_BOOT_INFO_REG0-7) hold the status information of the secure boot process for communication with the Host application. The definitions of the register fields (part of HSM_SOC_CTRL address space) are dependent on the implementation of the secure boot flow. These registers captures the output from the SoC system control module to the HSM. Reset type: PORESETn

3.13.2.59 SOC_SECURE_BOOT_INFO_REG2 Register (Offset = 50Ch) [Reset = 00000000h]

SOC_SECURE_BOOT_INFO_REG2 is shown in Figure 3-86 and described in Table 3-80.

Return to the Summary Table.

Communicate from CPU1 to HSM during secure Boot

Figure 3-86. SOC_SECURE_BOOT_INFO_REG2 Register

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

Table 3-80. SOC_SECURE_BOOT_INFO_REG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STATUS	R/W	0h	These set of SW registers (SOC_SECURE_BOOT_INFO_REG0-7) hold the status information of the secure boot process for communication with the Host application. The definitions of the register fields (part of HSM_SOC_CTRL address space) are dependent on the implementation of the secure boot flow. These registers captures the output from the SoC system control module to the HSM. Reset type: PORESETn

3.13.2.60 SOC_SECURE_BOOT_INFO_REG3 Register (Offset = 510h) [Reset = 00000000h]

SOC_SECURE_BOOT_INFO_REG3 is shown in Figure 3-87 and described in Table 3-81.

Return to the Summary Table.

Communicate from CPU1 to HSM during secure Boot

Figure 3-87. SOC_SECURE_BOOT_INFO_REG3 Register

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

Table 3-81. SOC_SECURE_BOOT_INFO_REG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STATUS	R/W	0h	These set of SW registers (SOC_SECURE_BOOT_INFO_REG0-7) hold the status information of the secure boot process for communication with the Host application. The definitions of the register fields (part of HSM_SOC_CTRL address space) are dependent on the implementation of the secure boot flow. These registers captures the output from the SoC system control module to the HSM. Reset type: PORESETn

3.13.2.61 SOC_SECURE_BOOT_INFO_REG4 Register (Offset = 514h) [Reset = 00000000h]

SOC_SECURE_BOOT_INFO_REG4 is shown in Figure 3-88 and described in Table 3-82.

Return to the Summary Table.

Communicate from CPU1 to HSM during secure Boot

Figure 3-88. SOC_SECURE_BOOT_INFO_REG4 Register

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

Table 3-82. SOC_SECURE_BOOT_INFO_REG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STATUS	R/W	0h	These set of SW registers (SOC_SECURE_BOOT_INFO_REG0-7) hold the status information of the secure boot process for communication with the Host application. The definitions of the register fields (part of HSM_SOC_CTRL address space) are dependent on the implementation of the secure boot flow. These registers captures the output from the SoC system control module to the HSM. Reset type: PORESETn

3.13.2.62 SOC_SECURE_BOOT_INFO_REG5 Register (Offset = 518h) [Reset = 00000000h]

SOC_SECURE_BOOT_INFO_REG5 is shown in Figure 3-89 and described in Table 3-83.

Return to the Summary Table.

Communicate from CPU1 to HSM during secure Boot

Figure 3-89. SOC_SECURE_BOOT_INFO_REG5 Register

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

Table 3-83. SOC_SECURE_BOOT_INFO_REG5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STATUS	R/W	0h	These set of SW registers (SOC_SECURE_BOOT_INFO_REG0-7) hold the status information of the secure boot process for communication with the Host application. The definitions of the register fields (part of HSM_SOC_CTRL address space) are dependent on the implementation of the secure boot flow. These registers captures the output from the SoC system control module to the HSM. Reset type: PORESETn

3.13.2.63 SOC_SECURE_BOOT_INFO_REG6 Register (Offset = 51Ch) [Reset = 00000000h]

SOC_SECURE_BOOT_INFO_REG6 is shown in Figure 3-90 and described in Table 3-84.

Return to the Summary Table.

Communicate from CPU1 to HSM during secure Boot

Figure 3-90. SOC_SECURE_BOOT_INFO_REG6 Register

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

Table 3-84. SOC_SECURE_BOOT_INFO_REG6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STATUS	R/W	0h	These set of SW registers (SOC_SECURE_BOOT_INFO_REG0-7) hold the status information of the secure boot process for communication with the Host application. The definitions of the register fields (part of HSM_SOC_CTRL address space) are dependent on the implementation of the secure boot flow. These registers captures the output from the SoC system control module to the HSM. Reset type: PORESETn

3.13.2.64 SOC_SECURE_BOOT_INFO_REG7 Register (Offset = 520h) [Reset = 00000000h]

SOC_SECURE_BOOT_INFO_REG7 is shown in Figure 3-91 and described in Table 3-85.

Return to the Summary Table.

Communicate from CPU1 to HSM during secure Boot

Figure 3-91. SOC_SECURE_BOOT_INFO_REG7 Register

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

Table 3-85. SOC_SECURE_BOOT_INFO_REG7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STATUS	R/W	0h	These set of SW registers (SOC_SECURE_BOOT_INFO_REG0-7) hold the status information of the secure boot process for communication with the Host application. The definitions of the register fields (part of HSM_SOC_CTRL address space) are dependent on the implementation of the secure boot flow. These registers captures the output from the SoC system control module to the HSM. Reset type: PORESETn

3.13.2.65 CLKCFGLOCK1 Register (Offset = 524h) [Reset = 00000000h]

CLKCFGLOCK1 is shown in Figure 3-92 and described in Table 3-86.

Return to the Summary Table.

Lock bit for CLKCFG registers
Notes:
[1] Any bit in this register, once set can only be cleared through a CPU1.SYSRSn. Write of 0 to any bit of this register has no effect
[2] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed

Figure 3-92. CLKCFGLOCK1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED		XCLKOUTDIVSEL	MCANCLKDIVSEL	HSMCLKDIVSEL	ETHERCATCLKCTL	EXTRFLTDET	XTALCR
R-0-0h		R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
RESERVED	CLBCLKCTL	PERCLKDIVSEL	RESERVED	SYSCLKDIVSEL	RESERVED	RESERVED	RESERVED
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
RESERVED	SYSPLLMULT	RESERVED	RESERVED	SYSPLLCTL1	CLKSRCCTL3	CLKSRCCTL2	CLKSRCCTL1
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 3-86. CLKCFGLOCK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R-0	0h	Reserved
21	XCLKOUTDIVSEL	R/WSonce	0h	Lock bit for XCLKOUTDIVSEL register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPU1.SYSRSn
20	MCANCLKDIVSEL	R/WSonce	0h	Lock bit for MCANCLKDIVSEL register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPU1.SYSRSn
19	HSMCLKDIVSEL	R/WSonce	0h	Lock bit for HSMCLKDIVSEL register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPU1.SYSRSn
18	ETHERCATCLKCTL	R/WSonce	0h	Lock bit for ETHERCATCLKCTL register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPU1.SYSRSn
17	EXTRFLTDET	R/WSonce	0h	Lock bit for EXTRFLTDET register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPU1.SYSRSn

Table 3-86. CLKCFGLOCK1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	XTALCR	R/WSonce	0h	Common Lock bit for XTALCR & XTAL CR2 register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPU1.SYSRSn
15	RESERVED	R/WSonce	0h	Reserved
14	CLBCLKCTL	R/WSonce	0h	Lock bit for CLBCLKCTL register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPU1.SYSRSn
13	PERCLKDIVSEL	R/WSonce	0h	Lock bit for PERCLKDIVSEL register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPU1.SYSRSn
12	RESERVED	R/WSonce	0h	Reserved
11	SYSCLKDIVSEL	R/WSonce	0h	Lock bit for SYSCLKDIVSEL register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPU1.SYSRSn
10	RESERVED	R/WSonce	0h	Reserved
9	RESERVED	R/WSonce	0h	Reserved
8	RESERVED	R/WSonce	0h	Reserved
7	RESERVED	R/WSonce	0h	Reserved
6	SYSPLLMULT	R/WSonce	0h	Lock bit for SYSPLLMULT register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPU1.SYSRSn
5	RESERVED	R/WSonce	0h	Reserved
4	RESERVED	R/WSonce	0h	Reserved
3	SYSPLLCTL1	R/WSonce	0h	Lock bit for SYSPLLCTL1 register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPU1.SYSRSn
2	CLKSRCCTL3	R/WSonce	0h	Lock bit for CLKSRCCTL3 register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPU1.SYSRSn
1	CLKSRCCTL2	R/WSonce	0h	Lock bit for CLKSRCCTL2 register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPU1.SYSRSn
0	CLKSRCCTL1	R/WSonce	0h	Lock bit for CLKSRCCTL1 register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPU1.SYSRSn

3.13.2.66 CLKSRCCTL1 Register (Offset = 530h) [Reset = 00000000h]

CLKSRCCTL1 is shown in Figure 3-93 and described in Table 3-87.

Return to the Summary Table.

Clock Source Control register-1
This memory mapped register requires a delay of 69 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 69 NOP instructions.

Figure 3-93. CLKSRCCTL1 Register

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

Table 3-87. CLKSRCCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R-0	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R-0	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R-0	0h	Reserved

Table 3-87. CLKSRCCTL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	OSCCLKSRCSEL	R/W	0h	Oscillator Clock Source Select Bit: This bit selects the source for OSCCLK. 00 = INTOSC2 (default on reset) 01 = External Oscillator (XTAL) 10 = INTOSC1 11 = reserved (default to INTOSC1) At power-up or after an XRSn, INTOSC2 is selected by default. Whenever the user changes the clock source using these bits, the SYSPLLMULT[13:0] register will be forced to zero and the PLL will be bypassed and powered down. This prevents potential PLL overshoot. The user will then have to write to the SYSPLLMULT register to configure the appropriate multiplier. The user must wait 10 OSCCLK cycles before writing to SYSPLLMULT or disabling the previous clock source to allow the change to complete.. Notes: [1] INTOSC1 is recommended to be used only after missing clock detection. If user wants to re-lock the PLL with INTOSC1 (the back-up clock source) after missing clock is detected, he can do a MCLKCLR and lock the PLL. Reset type: XRSn

3.13.2.67 CLKSRCCTL2 Register (Offset = 534h) [Reset = 00000000h]

CLKSRCCTL2 is shown in Figure 3-94 and described in Table 3-88.

Return to the Summary Table.

Clock Source Control register-2
This memory mapped register requires a delay of 69 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 69 NOP instructions.

Figure 3-94. CLKSRCCTL2 Register

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

Table 3-88. CLKSRCCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R-0	0h	Reserved
21-20	MCANFBCLKSEL	R/W	0h	MCAN Bit Clock Source Select Bit: 00 = PERx.SYSCLK 01 = AUXPLLRAWCLK (Reserved) 10 = AUXCLKIN 11 = PLLCLK Missing clock detect circuit doesnt have any impact on these bits. Reset type: XRSn
19-18	MCANEBCLKSEL	R/W	0h	MCAN Bit Clock Source Select Bit: 00 = PERx.SYSCLK 01 = AUXPLLRAWCLK (Reserved) 10 = AUXCLKIN 11 = PLLCLK Missing clock detect circuit doesnt have any impact on these bits. Reset type: XRSn
17-16	MCANDBCLKSEL	R/W	0h	MCAN Bit Clock Source Select Bit: 00 = PERx.SYSCLK 01 = AUXPLLRAWCLK (Reserved) 10 = AUXCLKIN 11 = PLLCLK Missing clock detect circuit doesnt have any impact on these bits. Reset type: XRSn
15-14	MCANCBCLKSEL	R/W	0h	MCAN Bit Clock Source Select Bit: 00 = PERx.SYSCLK 01 = AUXPLLRAWCLK (Reserved) 10 = AUXCLKIN 11 = PLLCLK Missing clock detect circuit doesnt have any impact on these bits. Reset type: XRSn

Table 3-88. CLKSRCCTL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-12	MCANBBCLKSEL	R/W	0h	MCAN Bit Clock Source Select Bit: 00 = PERx.SYSCLK 01 = AUXPLLRAWCLK (Reserved) 10 = AUXCLKIN 11 = PLLCLK Missing clock detect circuit doesnt have any impact on these bits. Reset type: XRSn
11-10	MCANABCLKSEL	R/W	0h	MCAN Bit Clock Source Select Bit: 00 = PERx.SYSCLK 01 = AUXPLLRAWCLK (Reserved) 10 = AUXCLKIN 11 = PLLCLK Missing clock detect circuit doesnt have any impact on these bits. Reset type: XRSn
9-8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3-2	RESERVED	R/W	0h	Reserved
1-0	RESERVED	R/W	0h	Reserved

3.13.2.68 CLKSRCCTL3 Register (Offset = 538h) [Reset = 00000000h]

CLKSRCCTL3 is shown in Figure 3-95 and described in Table 3-89.

Return to the Summary Table.

Clock Source Control register-3
This memory mapped register requires a delay of 69 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 69 NOP instructions.

Figure 3-95. CLKSRCCTL3 Register

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

Table 3-89. CLKSRCCTL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R-0	0h	Reserved
4-0	XCLKOUTSEL	R/W	0h	00000 = PLLSYSCLK (default on reset) 00001 = CPU1.CLOCK 00010 = CPU2.CLOCK 00011 = CPU3.CLOCK 00100 = Reserved (AUXPLLCLK - After the Bypass Mux) 00101 = INTOSC1 00110 = INTOSC2 00111 = XTAL OSC o/p clock 01000 = Reserved (CMCLK) 01001 = PUMPOSC0 (from no-wrapper0) 01010 = SYSAPLL.CLK_AUX 01011 = Reserved (AUXAPLL.CLK_AUX) 01100 = Reserved (AUXPLLRAWCLK) 01101 = PUMPOSC1 (from FLC1) 01110 = PUMPOSC2 (from FLC2) 01111 = PLLRAWCLK 10000 = PLLCLK (After the Bypass Mux) 10001 = Reserved (CPU4.CLOCK) 10010 = Reserved (CPU5.CLOCK) 10011 = Reserved (CPU6.CLOCK) 10100 = Reserved 10101 = Reserved 10110 = Reserved 10111 = Reserved 11000 = Reserved 11001 = Reserved 11010 = Reserved 11011 = Reserved 11100 = Reserved Reset type: XRSn

3.13.2.69 SYSPLLCTL1 Register (Offset = 53Ch) [Reset = 00000000h]

SYSPLLCTL1 is shown in Figure 3-96 and described in Table 3-90.

Return to the Summary Table.

SYSPLL Control register-1
This memory mapped register requires a delay of 69 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 69 NOP instructions.

Figure 3-96. SYSPLLCTL1 Register

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

Table 3-90. SYSPLLCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	PLLCLKEN	R/W	0h	SYSPLL bypassed or included in the PLLSYSCLK path: This bit decides if the SYSPLL is bypassed when PLLSYSCLK is generated 1 = PLLSYSCLK is fed from the SYSPLL clock output. Users need to make sure that the PLL is locked before enabling this clock to the system. 0 = SYSPLL is bypassed. Clock to system is direct feed from OSCCLK Reset type: XRSn
0	PLLEN	R/W	0h	SYSPLL enabled or disabled: This bit decides if the SYSPLL is enabled or not 1 = SYSPLL is enabled 0 = SYSPLL is powered off. Clock to system is direct feed from OSCCLK Reset type: XRSn

3.13.2.70 SYSPLLMULT Register (Offset = 548h) [Reset = 00000000h]

SYSPLLMULT is shown in Figure 3-97 and described in Table 3-91.

Return to the Summary Table.

SYSPLL Multiplier register
NOTE: FMULT and IMULT fields must be written at the same time for correct PLL operation.
This memory mapped register requires a delay of 69 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 69 NOP instructions.

Figure 3-97. SYSPLLMULT Register

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

Table 3-91. SYSPLLMULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R-0	0h	Reserved
28-24	REFDIV	R/W	0h	SYSPLL Reference Clock Divider PLL Reference Divider = REFDIV + 1 Reset type: XRSn
23-21	RESERVED	R-0	0h	Reserved
20-16	ODIV	R/W	0h	SYSPLL Output Clock Divider PLL Output Divider = ODIV + 1 Reset type: XRSn
15-14	RESERVED	R-0	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R-0	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-0	IMULT	R/W	0h	SYSPLL Integer Multiplier: For 00000000 Fout = Fref (PLLBYPASS) Integer Multiplier = 1 00000001 Integer Multiplier = 1 00000010 Integer Multiplier = 2 00000011 Integer Multiplier = 3 ..... 01111111 Integer Multipler = 127 ..... 11111111 Integer Multipler = 255 Note for APLL Multiplier values from 0-3 are invalid, internally those will be treated to 4. Reset type: XRSn

3.13.2.71 SYSPLLSTS Register (Offset = 54Ch) [Reset = 00000030h]

SYSPLLSTS is shown in Figure 3-98 and described in Table 3-92.

Return to the Summary Table.

SYSPLL Status register

Figure 3-98. SYSPLLSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		RESERVED	RESERVED	RESERVED	RESERVED	SLIPS_NOTSUPPORTED	LOCKS
R-0-0h		R-1h	R-1h	W1C-0h	R-0h	R-0h	R-0h

Table 3-92. SYSPLLSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	RESERVED	R	1h	Reserved
4	RESERVED	R	1h	Reserved
3	RESERVED	W1C	0h	Reserved
2	RESERVED	R	0h	Reserved
1	SLIPS_NOTSUPPORTED	R	0h	RESERVED: This bit is reserved and the value read should be ignored. TI recommends using DCC to evaluate SYSPLL Slip status. Refer to InitSysPll() or SysCtl_setClock() functions inside the latest example software from C2000Ware for checking SYSPLL Slip status using DCC. Reset type: XRSn
0	LOCKS	R	0h	SYSPLL Lock Status Bit: This bit indicates whether the SYSPLL is locked or not 0 = SYSPLL is not yet locked 1 = SYSPLL is locked Reset type: XRSn

3.13.2.72 SYSCLKDIVSEL Register (Offset = 564h) [Reset = 00000000h]

SYSCLKDIVSEL is shown in Figure 3-99 and described in Table 3-93.

Return to the Summary Table.

System Clock Divider Select register
This memory mapped register requires a delay of 69 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 69 NOP instructions.

Figure 3-99. SYSCLKDIVSEL Register

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

Table 3-93. SYSCLKDIVSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R-0	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R-0	0h	Reserved
5-0	PLLSYSCLKDIV	R/W	0h	PLLSYSCLK Divide Select: This bit selects the divider setting for the PLLSYSCLK. 000000 = /1 (Default) 000001 = /2 000010 = /3 000011 = /4 000100 = /5 ..... 111111 = /64 Reset type: XRSn

3.13.2.73 PERCLKDIVSEL Register (Offset = 56Ch) [Reset = 00000911h]

PERCLKDIVSEL is shown in Figure 3-100 and described in Table 3-94.

Return to the Summary Table.

Peripheral Clock Divider Select register

Figure 3-100. PERCLKDIVSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED			LINBCLKDIV		RESERVED	LINACLKDIV	
R-0-0h			R/W-1h		R-0-0h	R/W-1h	
7	6	5	4	3	2	1	0
RESERVED	RESERVED	EMIF1CLKDIV		RESERVED		EPWMCLKDIV	
R-0-0h	R/W-0h	R/W-1h		R/W-0h		R/W-1h	

Table 3-94. PERCLKDIVSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R-0	0h	Reserved
12-11	LINBCLKDIV	R/W	1h	LINB Clock Divide Select: This bit selects whether the LINB module run with a /1 or /2 clock. 00: /1 of SYSCLK is selected 01: /2 of SYSCLK is selected 10: /4 of SYSCLK is selected 11: Reserved Reset type: CPU1.SYSRSn
10	RESERVED	R-0	0h	Reserved
9-8	LINACLKDIV	R/W	1h	LINA Clock Divide Select: This bit selects whether the LINA module run with a /1 or /2 clock. 00: /1 of SYSCLK is selected 01: /2 of SYSCLK is selected 10: /4 of SYSCLK is selected 11: Reserved Reset type: CPU1.SYSRSn
7	RESERVED	R-0	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5-4	EMIF1CLKDIV	R/W	1h	EMIF1 Clock Divide Select: This bit selects whether the EMIF1 module run with a /1 /2, or /4 clock. For Dual core device 0: /1 of PLLSYSCLK is selected 1: /2 of PLLSYSCLK is selected 2: /4 of PLLSYSCLK is selected 3: Reserved Reset type: CPU1.SYSRSn
3-2	RESERVED	R/W	0h	Reserved

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

Bit	Field	Type	Reset	Description
1-0	EPWMCLKDIV	R/W	1h	EPWM Clock Divide Select: This bit selects whether the EPWM modules run with a /1 or /2 clock. This divider sits in front of the PLLSYSCLK x0 = /1 of SYSCLK x1 = /2 of SYSLCK Note: Refer to the EPWM User Guide for maximum EPWM Frequency Reset type: CPU1.SYSRSn

3.13.2.74 XCLKOUTDIVSEL Register (Offset = 570h) [Reset = 00000003h]

XCLKOUTDIVSEL is shown in Figure 3-101 and described in Table 3-95.

Return to the Summary Table.

XCLKOUT Divider Select register
This memory mapped register requires a delay of 69 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 69 NOP instructions.

Figure 3-101. XCLKOUTDIVSEL Register

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

Table 3-95. XCLKOUTDIVSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1-0	XCLKOUTDIV	R/W	3h	XCLKOUT Divide Select: This bit selects the divider setting for the XCLKOUT. 00 = /1 01 = /2 10 = /4 11 = /8 (default on reset) Reset type: CPU1.SYSRSn

3.13.2.75 HSMCLKDIVSEL Register (Offset = 574h) [Reset = 00000001h]

HSMCLKDIVSEL is shown in Figure 3-102 and described in Table 3-96.

Return to the Summary Table.

HSM SYSCLK Divider Select register

Figure 3-102. HSMCLKDIVSEL Register

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

Table 3-96. HSMCLKDIVSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R-0	0h	Reserved
4-0	HSMCLKDIV	R/W	1h	00001 = /2 00010 = /4 ... Rest = Should be treated as value 00001=/2 Reset type: XRSn

3.13.2.76 MCANCLKDIVSEL Register (Offset = 578h) [Reset = 2739CE73h]

MCANCLKDIVSEL is shown in Figure 3-103 and described in Table 3-97.

Return to the Summary Table.

MCAN Bit Clock Divider Select register

Figure 3-103. MCANCLKDIVSEL Register

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

Table 3-97. MCANCLKDIVSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R-0	0h	Reserved
29-25	MCANFCLKDIV	R/W	13h	00000 = /1 00001 = /2 ... 10010 = /19 10011 = /20 101xx = Rsvd 11xxx = Rsvd Reset type: XRSn
24-20	MCANECLKDIV	R/W	13h	00000 = /1 00001 = /2 ... 10010 = /19 10011 = /20 101xx = Rsvd 11xxx = Rsvd Reset type: XRSn
19-15	MCANDCLKDIV	R/W	13h	00000 = /1 00001 = /2 ... 10010 = /19 10011 = /20 101xx = Rsvd 11xxx = Rsvd Reset type: XRSn

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

Bit	Field	Type	Reset	Description
14-10	MCANCCLKDIV	R/W	13h	00000 = /1 00001 = /2 ... 10010 = /19 10011 = /20 101xx = Rsvd 11xxx = Rsvd Reset type: XRSn
9-5	MCANBCLKDIV	R/W	13h	00000 = /1 00001 = /2 ... 10010 = /19 10011 = /20 101xx = Rsvd 11xxx = Rsvd Reset type: XRSn
4-0	MCANACLKDIV	R/W	13h	00000 = /1 00001 = /2 ... 10010 = /19 10011 = /20 101xx = Rsvd 11xxx = Rsvd Reset type: XRSn

3.13.2.77 CLBCLKCTL Register (Offset = 57Ch) [Reset = 00000007h]

CLBCLKCTL is shown in Figure 3-104 and described in Table 3-98.

Return to the Summary Table.

CLB Clocking Control Register

Figure 3-104. CLBCLKCTL Register

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

Table 3-98. CLBCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	CLKMODECLB6	R/W	0h	0 : CLB6 is synchronous to SYSCLK 1 : CLB6 runs of asynchronous clock Reset type: SYSRSn
20	CLKMODECLB5	R/W	0h	0 : CLB5 is synchronous to SYSCLK 1 : CLB5 runs of asynchronous clock Reset type: SYSRSn
19	CLKMODECLB4	R/W	0h	0 : CLB4 is synchronous to SYSCLK 1 : CLB4 runs of asynchronous clock Reset type: SYSRSn
18	CLKMODECLB3	R/W	0h	0 : CLB3 is synchronous to SYSCLK 1 : CLB3 runs of asynchronous clock Reset type: SYSRSn
17	CLKMODECLB2	R/W	0h	0 : CLB2 is synchronous to SYSCLK 1 : CLB2 runs of asynchronous clock Reset type: SYSRSn
16	CLKMODECLB1	R/W	0h	0 : CLB1 is synchronous to SYSCLK 1 : CLB1 runs of asynchronous clock Reset type: SYSRSn
15-5	RESERVED	R-0	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R-0	0h	Reserved
2-0	RESERVED	R/W	7h	Reserved

3.13.2.78 MCDCR Register (Offset = 584h) [Reset = 00000000h]

MCDCR is shown in Figure 3-105 and described in Table 3-99.

Return to the Summary Table.

Missing Clock Detect Control Register

Figure 3-105. MCDCR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	EXTR_FAULT_MCD_EN	EXTR_FAULTSCLR	EXTR_FAULTS	RESERVED	RESERVED
R-0-0h	R/W-0h	R/W-0h	R/W-0h	R-0/W1S-0h	R-0h	R/W-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	SYSREF_LOST_MCD_EN	SYSREF_LOSTSCLR	SYSREF_LOSTS	OSCOFF	MCLKOFF	MCLKCLR	MCLKSTS
R-0h	R/W-0h	R-0/W1S-0h	R-0h	R/W-0h	R/W-0h	R-0/W1S-0h	R-0h

Table 3-99. MCDCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R-0	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	EXTR_FAULT_MCD_EN	R/W	0h	Control to add 'EXTR FAULTt' as cause for MCD 0 = 'EXTR FAULT' does not affect MCD. 1 = Upon 'EXTR FAULT' MCD is asserted. Reset type: XRSn
11	EXTR_FAULTSCLR	R-0/W1S	0h	Clears the EXTR_FAULTS from MCDCR which is root for MCD trigger. 0 = No effect on present state of the EXTR_FAULTS 1 = Clears the EXTR_FAULTS bit to '0'. Bit clears itself after clear pulse to EXTR_FAULTS. Read always gives '0'. Reset type: XRSn
10	EXTR_FAULTS	R	0h	External Resistor fault status Bit: This bit indicates whether there is a critical fault in the external resistor connected to the device 0 = 'EXTR fault' event has not occurred. 1 = 'EXTR fault' event has occurred. Reset type: XRSn
9	RESERVED	R/W	0h	Reserved
8	RESERVED	R-0/W1S	0h	Reserved
7	RESERVED	R	0h	Reserved
6	SYSREF_LOST_MCD_EN	R/W	0h	Control to add 'PLL reference clock lost' as cause for MCD 0 = 'PLL reference clock Lost' does not affect MCD. 1 = Upon 'PLL reference clock Lost' MCD is asserted. Reset type: XRSn

Table 3-99. MCDCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	SYSREF_LOSTSCLR	R-0/W1S	0h	Clears the REF_LOST_STS from PLLSTS which is root for MCD trigger. 0 = No effect on present state of the REF_LOST_STS 1 = Clears the REF_LOST_STS bit to '0'. Bit clears itself after clear pulse to REF_LOST_STS. Read always gives '0'. Reset type: XRSn
4	SYSREF_LOSTS	R	0h	SYSPLL 'Reference Lost' Status Bit: This bit indicates whether the SYSPLL is out of lock range 0 = 'Reference Lost' event has not occurred. 1 = 'Reference Lost' event has occurred. Reset type: XRSn
3	OSCOFF	R/W	0h	Oscillator Clock Disconnect from MCD Bit: 0 = OSCCLK Connected to OSCCLK Counter in MCD module 1 = OSCCLK Disconnected to OSCCLK Counter in MCD module Reset type: XRSn
2	MCLKOFF	R/W	0h	Missing Clock Detect Off Bit: 0 = Missing Clock Detect Circuit Enabled 1 = Missing Clock Detect Circuit Disabled Reset type: XRSn
1	MCLKCLR	R-0/W1S	0h	Missing Clock Clear Bit: Write 1' to this bit to clear MCLKSTS bit and reset the missing clock detect circuit.' Reset type: XRSn
0	MCLKSTS	R	0h	Missing Clock Status Bit: 0 = OSCCLK Is OK 1 = OSCCLK Detected Missing, CLOCKFAILn Generated Reset type: XRSn

3.13.2.79 X1CNT Register (Offset = 588h) [Reset = 00000000h]

X1CNT is shown in Figure 3-106 and described in Table 3-100.

Return to the Summary Table.

10-bit Counter on X1 Clock

Figure 3-106. X1CNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															CLR
R-0-0h															R-0/W1C-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					X1CNT										
R-0-0h					R-0h										

Table 3-100. X1CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R-0	0h	Reserved
16	CLR	R-0/W1C	0h	X1 Counter clear: A write of '1' to this bit field clears the X1CNT and makes it count from 0x0 again (provided X1 clock is ticking). Writes of '0' are ignore to this bit field Reset type: XRSn
15-11	RESERVED	R-0	0h	Reserved
10-0	X1CNT	R	0h	X1 Counter: - This counter increments on every X1 CLOCKs positive-edge. - Once it reaches the values of 0x7ff, it freezes - Before switching from INTOSC2 to X1, application must check this counter and make sure that it has saturated. This will ensure that the Crystal connected to X1/X2 is oscillating. Reset type: XRSn

3.13.2.80 XTALCR Register (Offset = 58Ch) [Reset = 00000001h]

XTALCR is shown in Figure 3-107 and described in Table 3-101.

Return to the Summary Table.

XTAL Control Register
This memory mapped register requires a delay of 69 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 69 NOP instructions.

Figure 3-107. XTALCR Register

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

Table 3-101. XTALCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	SE	R/W	0h	Configures XTAL oscillator in single-ended or Crystal mode when XTAL oscillator is powered up(i.e. OSCOFF = 0) 0 XTAL oscillator in Crystal mode 1 XTAL oscilator in single-ended mode (through X1) Reset type: XRSn
0	OSCOFF	R/W	1h	This bit if '1', powers-down the XTAL oscillator macro and hence doesn't let X2 to be driven by the XTAL oscillator. If a crystal is connected to X1/X2, user needs to first clear this bit, wait for the oscillator to power up (using X1CNT) and then only switch the clock source to X1/X2 Reset type: XRSn

3.13.2.81 XTALCR2 Register (Offset = 59Ch) [Reset = 00000003h]

XTALCR2 is shown in Figure 3-108 and described in Table 3-102.

Return to the Summary Table.

XTAL Control Register for pad init

Figure 3-108. XTALCR2 Register

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

Table 3-102. XTALCR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Reserved
15-3	RESERVED	R-0	0h	Reserved
2	FEN	R/W	0h	Configures XTAL oscillator pad initilisation. 0 : XOSC pads are not driven through GPIO connection. 1 : XOSC pads are driven through connected GPIO as per XIF & XOF values. This register has effect only when XOSC is OFF (no SE , no XTAL mode). If this register is set during XOSC off state (XOSCOFF=1 & SE=0) then upon change of these controls this bit gets reset and rearmed. Reset type: XRSn
1	XOF	R/W	1h	Polarity selection to initialise XO /X2 pad of the XOSC before start-up This value shall be deposited on the pad before XOSC started (XOSCOFF=1) If FEN=0 or XOSC is in XTAL or SE mode then this value will not be applied to the pad. Reset type: XRSn
0	XIF	R/W	1h	Polarity selection to initialise XI /X1 pad of the XOSC before start-up This value shall be deposited on the pad before XOSC started (XOSCOFF=1) If FEN=0 or XOSC is in XTAL or SE mode then this value will not be applied to the pad. Reset type: XRSn

3.13.2.82 ETHERCATCLKCTL Register (Offset = 5A8h) [Reset = 0000000Eh]

ETHERCATCLKCTL is shown in Figure 3-109 and described in Table 3-103.

Return to the Summary Table.

EtherCAT Clock Control

Figure 3-109. ETHERCATCLKCTL Register

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

Table 3-103. ETHERCATCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	Reserved
8	PHYCLKEN	R/W	0h	0 : etherCAT phy clock disabled 1 : etherCAT phy clock enabled Reset type: XRSn
7-4	RESERVED	R-0	0h	Reserved
3-1	ECATDIV	R/W	7h	000: /1 001: /2 010: /3 011: /4 100: /5 101: /6 110: /7 111: /8 Reset type: XRSn
0	RESERVED	R/W	0h	Reserved

3.13.2.83 ETHERCATCTL Register (Offset = 5ACh) [Reset = 00000000h]

ETHERCATCTL is shown in Figure 3-110 and described in Table 3-104.

Return to the Summary Table.

ETHERCAT control register.

Figure 3-110. ETHERCATCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							I2CLOOPBACK
R-0-0h							R/W-0h

Table 3-104. ETHERCATCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	I2CLOOPBACK	R/W	0h	ETHERCAT I2C loopback enable Bit: 0: I2C port of etherCAT is not looped back to I2C_A 1: I2C port of etherCAT is looped back to I2C_A Reset type: XRSn

3.13.2.84 SYNCBUSY Register (Offset = 5B0h) [Reset = 00000000h]

SYNCBUSY is shown in Figure 3-111 and described in Table 3-105.

Return to the Summary Table.

Pulse Transfer Sync Busy Status register

Figure 3-111. SYNCBUSY Register

31	30	29	28	27	26	25	24
CPU2TMR2CTL	CPU1TMR2CTL	CPU3TMR2CTL	CLKSRCCTL3	CLKSRCCTL2	CLKSRCCTL1	XTALCR	XCLKOUTDIVSEL
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
SYSPLLMULT	SYSPLLCTL1	SYSCLKDIVSEL	PERCLKDIVSEL	ETHERCATCLKCTL	CLBCLKCTL	RESERVED	MCANCLKDIVSEL
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							BUSY
R-0-0h							R-0h

Table 3-105. SYNCBUSY Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CPU2TMR2CTL	R	0h	This status bit indicates write to the register is in progress 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn
30	CPU1TMR2CTL	R	0h	This status bit indicates write to the register is in progress 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn
29	CPU3TMR2CTL	R	0h	This status bit indicates write to the register is in progress 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn
28	CLKSRCCTL3	R	0h	This status bit indicates write to the register is in progress 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn
27	CLKSRCCTL2	R	0h	This status bit indicates write to the register is in progress 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn
26	CLKSRCCTL1	R	0h	This status bit indicates write to the register is in progress 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn
25	XTALCR	R	0h	This status bit indicates write to the register is in progress 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn

Table 3-105. SYNCBUSY Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	XCLKOUTDIVSEL	R	0h	This status bit indicates write to the register is in progress 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn
23	SYSPLLMULT	R	0h	This status bit indicates write to the register is in progress 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn
22	SYSPLLCTL1	R	0h	This status bit indicates write to the register is in progress 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn
21	SYSCLKDIVSEL	R	0h	This status bit indicates write to the register is in progress 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn
20	PERCLKDIVSEL	R	0h	This status bit indicates write to the register is in progress 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn
19	ETHERCATCLKCTL	R	0h	This status bit indicates write to the register is in progress 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn
18	CLBCLKCTL	R	0h	This status bit indicates write to the register is in progress 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn
17	RESERVED	R	0h	Reserved
16	MCANCLKDIVSEL	R	0h	This status bit indicates write to the register is in progress 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn
15-1	RESERVED	R-0	0h	Reserved
0	BUSY	R	0h	This status bit indicates write to any of the following registers (OR_REDUCE) is in progress or not. MCANCLKDIVSEL,CLBCLKCTL,ETHERCATCLKCTL,PERCLKDIVSEL,SYSCLKDIVSEL,SYSPLLCTL1,SYSPLLMULT,XCLKOUTDIVSEL,XTALCR,CLKSRCCTL1,CLKSRCCTL2,CLKSRCCTL3,CPU3TMR2CTL,CPU1TMR2CTL,CPU2TMR2CTL 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn

3.13.2.85 ESMXRSNCTL Register (Offset = 5C0h) [Reset = 00010001h]

ESMXRSNCTL is shown in Figure 3-112 and described in Table 3-106.

Return to the Summary Table.

Enable ESM reset outputs for XRSn

Figure 3-112. ESMXRSNCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							ESMRESET
R-0-0h							R/W-1h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		ESMCPU3HIPRIWD	ESMCPU3CRITICAL	ESMCPU2HIPRIWD	ESMCPU2CRITICAL	RESERVED	ESMCPU1CRITICAL
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h

Table 3-106. ESMXRSNCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R-0	0h	Reserved
16	ESMRESET	R/W	1h	If this bit is set, ESM output will be enabled to cause respective reset Reset type: PORESETn
15-6	RESERVED	R-0	0h	Reserved
5	ESMCPU3HIPRIWD	R/W	0h	If this bit is set, ESM output will be enabled to cause XRSN Reset type: PORESETn
4	ESMCPU3CRITICAL	R/W	0h	If this bit is set, ESM output will be enabled to cause XRSN Reset type: PORESETn
3	ESMCPU2HIPRIWD	R/W	0h	If this bit is set, ESM output will be enabled to cause XRSN Reset type: PORESETn
2	ESMCPU2CRITICAL	R/W	0h	If this bit is set, ESM output will be enabled to cause XRSN Reset type: PORESETn
1	RESERVED	R/W	0h	Reserved
0	ESMCPU1CRITICAL	R/W	1h	If this bit is set, respective ESM output will be enabled to cause XRSN Reset type: PORESETn

3.13.2.86 EPWM1 Register (Offset = 5C8h) [Reset = 000000C0h]

EPWM1 is shown in Figure 3-113 and described in Table 3-107.

Return to the Summary Table.

PER2SYSCONFIG - Peripheral System Configuration for EPWM1

Figure 3-113. EPWM1 Register

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

Table 3-107. EPWM1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.87 EPWM2 Register (Offset = 5CCh) [Reset = 000000C0h]

EPWM2 is shown in Figure 3-114 and described in Table 3-108.

Return to the Summary Table.

PER3SYSCONFIG - Peripheral System Configuration for EPWM2

Figure 3-114. EPWM2 Register

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

Table 3-108. EPWM2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.88 EPWM3 Register (Offset = 5D0h) [Reset = 000000C0h]

EPWM3 is shown in Figure 3-115 and described in Table 3-109.

Return to the Summary Table.

PER4SYSCONFIG - Peripheral System Configuration for EPWM3

Figure 3-115. EPWM3 Register

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

Table 3-109. EPWM3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.89 EPWM4 Register (Offset = 5D4h) [Reset = 000000C0h]

EPWM4 is shown in Figure 3-116 and described in Table 3-110.

Return to the Summary Table.

PER5SYSCONFIG - Peripheral System Configuration for EPWM4

Figure 3-116. EPWM4 Register

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

Table 3-110. EPWM4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.90 EPWM5 Register (Offset = 5D8h) [Reset = 000000C0h]

EPWM5 is shown in Figure 3-117 and described in Table 3-111.

Return to the Summary Table.

PER6SYSCONFIG - Peripheral System Configuration for EPWM5

Figure 3-117. EPWM5 Register

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

Table 3-111. EPWM5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.91 EPWM6 Register (Offset = 5DCh) [Reset = 000000C0h]

EPWM6 is shown in Figure 3-118 and described in Table 3-112.

Return to the Summary Table.

PER7SYSCONFIG - Peripheral System Configuration for EPWM6

Figure 3-118. EPWM6 Register

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

Table 3-112. EPWM6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.92 EPWM7 Register (Offset = 5E0h) [Reset = 000000C0h]

EPWM7 is shown in Figure 3-119 and described in Table 3-113.

Return to the Summary Table.

PER8SYSCONFIG - Peripheral System Configuration for EPWM7

Figure 3-119. EPWM7 Register

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

Table 3-113. EPWM7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.93 EPWM8 Register (Offset = 5E4h) [Reset = 000000C0h]

EPWM8 is shown in Figure 3-120 and described in Table 3-114.

Return to the Summary Table.

PER9SYSCONFIG - Peripheral System Configuration for EPWM8

Figure 3-120. EPWM8 Register

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

Table 3-114. EPWM8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.94 EPWM9 Register (Offset = 5E8h) [Reset = 000000C0h]

EPWM9 is shown in Figure 3-121 and described in Table 3-115.

Return to the Summary Table.

PER10SYSCONFIG - Peripheral System Configuration for EPWM9

Figure 3-121. EPWM9 Register

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

Table 3-115. EPWM9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.95 EPWM10 Register (Offset = 5ECh) [Reset = 000000C0h]

EPWM10 is shown in Figure 3-122 and described in Table 3-116.

Return to the Summary Table.

PER11SYSCONFIG - Peripheral System Configuration for EPWM10

Figure 3-122. EPWM10 Register

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

Table 3-116. EPWM10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.96 EPWM11 Register (Offset = 5F0h) [Reset = 000000C0h]

EPWM11 is shown in Figure 3-123 and described in Table 3-117.

Return to the Summary Table.

PER12SYSCONFIG - Peripheral System Configuration for EPWM11

Figure 3-123. EPWM11 Register

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

Table 3-117. EPWM11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.97 EPWM12 Register (Offset = 5F4h) [Reset = 000000C0h]

EPWM12 is shown in Figure 3-124 and described in Table 3-118.

Return to the Summary Table.

PER13SYSCONFIG - Peripheral System Configuration for EPWM12

Figure 3-124. EPWM12 Register

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

Table 3-118. EPWM12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.98 EPWM13 Register (Offset = 5F8h) [Reset = 000000C0h]

EPWM13 is shown in Figure 3-125 and described in Table 3-119.

Return to the Summary Table.

PER14SYSCONFIG - Peripheral System Configuration for EPWM13

Figure 3-125. EPWM13 Register

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

Table 3-119. EPWM13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.99 EPWM14 Register (Offset = 5FCh) [Reset = 000000C0h]

EPWM14 is shown in Figure 3-126 and described in Table 3-120.

Return to the Summary Table.

PER15SYSCONFIG - Peripheral System Configuration for EPWM14

Figure 3-126. EPWM14 Register

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

Table 3-120. EPWM14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.100 EPWM15 Register (Offset = 600h) [Reset = 000000C0h]

EPWM15 is shown in Figure 3-127 and described in Table 3-121.

Return to the Summary Table.

PER16SYSCONFIG - Peripheral System Configuration for EPWM15

Figure 3-127. EPWM15 Register

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

Table 3-121. EPWM15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.101 EPWM16 Register (Offset = 604h) [Reset = 000000C0h]

EPWM16 is shown in Figure 3-128 and described in Table 3-122.

Return to the Summary Table.

PER17SYSCONFIG - Peripheral System Configuration for EPWM16

Figure 3-128. EPWM16 Register

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

Table 3-122. EPWM16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.102 EPWM17 Register (Offset = 608h) [Reset = 000000C0h]

EPWM17 is shown in Figure 3-129 and described in Table 3-123.

Return to the Summary Table.

PER18SYSCONFIG - Peripheral System Configuration for EPWM17

Figure 3-129. EPWM17 Register

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

Table 3-123. EPWM17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.103 EPWM18 Register (Offset = 60Ch) [Reset = 000000C0h]

EPWM18 is shown in Figure 3-130 and described in Table 3-124.

Return to the Summary Table.

PER19SYSCONFIG - Peripheral System Configuration for EPWM18

Figure 3-130. EPWM18 Register

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

Table 3-124. EPWM18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.104 HRCAL0 Register (Offset = 614h) [Reset = 000000C0h]

HRCAL0 is shown in Figure 3-131 and described in Table 3-125.

Return to the Summary Table.

PER21SYSCONFIG - Peripheral System Configuration for HRCAL0

Figure 3-131. HRCAL0 Register

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

Table 3-125. HRCAL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.105 HRCAL1 Register (Offset = 618h) [Reset = 000000C0h]

HRCAL1 is shown in Figure 3-132 and described in Table 3-126.

Return to the Summary Table.

PER22SYSCONFIG - Peripheral System Configuration for HRCAL1

Figure 3-132. HRCAL1 Register

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

Table 3-126. HRCAL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.106 HRCAL2 Register (Offset = 61Ch) [Reset = 000000C0h]

HRCAL2 is shown in Figure 3-133 and described in Table 3-127.

Return to the Summary Table.

PER23SYSCONFIG - Peripheral System Configuration for HRCAL2

Figure 3-133. HRCAL2 Register

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

Table 3-127. HRCAL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.107 ECAP1 Register (Offset = 620h) [Reset = 000000C0h]

ECAP1 is shown in Figure 3-134 and described in Table 3-128.

Return to the Summary Table.

PER24SYSCONFIG - Peripheral System Configuration for ECAP1

Figure 3-134. ECAP1 Register

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

Table 3-128. ECAP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.108 ECAP2 Register (Offset = 624h) [Reset = 000000C0h]

ECAP2 is shown in Figure 3-135 and described in Table 3-129.

Return to the Summary Table.

PER25SYSCONFIG - Peripheral System Configuration for ECAP2

Figure 3-135. ECAP2 Register

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

Table 3-129. ECAP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.109 ECAP3 Register (Offset = 628h) [Reset = 000000C0h]

ECAP3 is shown in Figure 3-136 and described in Table 3-130.

Return to the Summary Table.

PER26SYSCONFIG - Peripheral System Configuration for ECAP3

Figure 3-136. ECAP3 Register

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

Table 3-130. ECAP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.110 ECAP4 Register (Offset = 62Ch) [Reset = 000000C0h]

ECAP4 is shown in Figure 3-137 and described in Table 3-131.

Return to the Summary Table.

PER27SYSCONFIG - Peripheral System Configuration for ECAP4

Figure 3-137. ECAP4 Register

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

Table 3-131. ECAP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.111 ECAP5 Register (Offset = 630h) [Reset = 000000C0h]

ECAP5 is shown in Figure 3-138 and described in Table 3-132.

Return to the Summary Table.

PER28SYSCONFIG - Peripheral System Configuration for ECAP5

Figure 3-138. ECAP5 Register

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

Table 3-132. ECAP5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.112 ECAP6 Register (Offset = 634h) [Reset = 000000C0h]

ECAP6 is shown in Figure 3-139 and described in Table 3-133.

Return to the Summary Table.

PER29SYSCONFIG - Peripheral System Configuration for ECAP6

Figure 3-139. ECAP6 Register

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

Table 3-133. ECAP6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.113 EQEP1 Register (Offset = 638h) [Reset = 000000C0h]

EQEP1 is shown in Figure 3-140 and described in Table 3-134.

Return to the Summary Table.

PER30SYSCONFIG - Peripheral System Configuration for EQEP1

Figure 3-140. EQEP1 Register

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

Table 3-134. EQEP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.114 EQEP2 Register (Offset = 63Ch) [Reset = 000000C0h]

EQEP2 is shown in Figure 3-141 and described in Table 3-135.

Return to the Summary Table.

PER31SYSCONFIG - Peripheral System Configuration for EQEP2

Figure 3-141. EQEP2 Register

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

Table 3-135. EQEP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.115 EQEP3 Register (Offset = 640h) [Reset = 000000C0h]

EQEP3 is shown in Figure 3-142 and described in Table 3-136.

Return to the Summary Table.

PER32SYSCONFIG - Peripheral System Configuration for EQEP3

Figure 3-142. EQEP3 Register

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

Table 3-136. EQEP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.116 EQEP4 Register (Offset = 644h) [Reset = 000000C0h]

EQEP4 is shown in Figure 3-143 and described in Table 3-137.

Return to the Summary Table.

PER33SYSCONFIG - Peripheral System Configuration for EQEP4

Figure 3-143. EQEP4 Register

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

Table 3-137. EQEP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.117 EQEP5 Register (Offset = 648h) [Reset = 000000C0h]

EQEP5 is shown in Figure 3-144 and described in Table 3-138.

Return to the Summary Table.

PER34SYSCONFIG - Peripheral System Configuration for EQEP5

Figure 3-144. EQEP5 Register

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

Table 3-138. EQEP5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.118 EQEP6 Register (Offset = 64Ch) [Reset = 000000C0h]

EQEP6 is shown in Figure 3-145 and described in Table 3-139.

Return to the Summary Table.

PER35SYSCONFIG - Peripheral System Configuration for EQEP6

Figure 3-145. EQEP6 Register

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

Table 3-139. EQEP6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.119 SDFM1 Register (Offset = 650h) [Reset = 000000C0h]

SDFM1 is shown in Figure 3-146 and described in Table 3-140.

Return to the Summary Table.

PER36SYSCONFIG - Peripheral System Configuration for SDFM1

Figure 3-146. SDFM1 Register

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

Table 3-140. SDFM1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.120 SDFM2 Register (Offset = 654h) [Reset = 000000C0h]

SDFM2 is shown in Figure 3-147 and described in Table 3-141.

Return to the Summary Table.

PER37SYSCONFIG - Peripheral System Configuration for SDFM2

Figure 3-147. SDFM2 Register

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

Table 3-141. SDFM2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.121 SDFM3 Register (Offset = 658h) [Reset = 000000C0h]

SDFM3 is shown in Figure 3-148 and described in Table 3-142.

Return to the Summary Table.

PER38SYSCONFIG - Peripheral System Configuration for SDFM3

Figure 3-148. SDFM3 Register

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

Table 3-142. SDFM3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.122 SDFM4 Register (Offset = 65Ch) [Reset = 000000C0h]

SDFM4 is shown in Figure 3-149 and described in Table 3-143.

Return to the Summary Table.

PER39SYSCONFIG - Peripheral System Configuration for SDFM4

Figure 3-149. SDFM4 Register

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

Table 3-143. SDFM4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.123 UARTA Register (Offset = 660h) [Reset = 000000C0h]

UARTA is shown in Figure 3-150 and described in Table 3-144.

Return to the Summary Table.

PER40SYSCONFIG - Peripheral System Configuration for UARTA

Figure 3-150. UARTA Register

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

Table 3-144. UARTA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.124 UARTB Register (Offset = 664h) [Reset = 000000C0h]

UARTB is shown in Figure 3-151 and described in Table 3-145.

Return to the Summary Table.

PER41SYSCONFIG - Peripheral System Configuration for UARTB

Figure 3-151. UARTB Register

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

Table 3-145. UARTB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.125 UARTC Register (Offset = 668h) [Reset = 000000C0h]

UARTC is shown in Figure 3-152 and described in Table 3-146.

Return to the Summary Table.

PER42SYSCONFIG - Peripheral System Configuration for UARTC

Figure 3-152. UARTC Register

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

Table 3-146. UARTC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.126 UARTD Register (Offset = 66Ch) [Reset = 000000C0h]

UARTD is shown in Figure 3-153 and described in Table 3-147.

Return to the Summary Table.

PER43SYSCONFIG - Peripheral System Configuration for UARTD

Figure 3-153. UARTD Register

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

Table 3-147. UARTD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.127 UARTE Register (Offset = 670h) [Reset = 000000C0h]

UARTE is shown in Figure 3-154 and described in Table 3-148.

Return to the Summary Table.

PER44SYSCONFIG - Peripheral System Configuration for UARTE

Figure 3-154. UARTE Register

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

Table 3-148. UARTE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.128 UARTF Register (Offset = 674h) [Reset = 000000C0h]

UARTF is shown in Figure 3-155 and described in Table 3-149.

Return to the Summary Table.

PER45SYSCONFIG - Peripheral System Configuration for UARTF

Figure 3-155. UARTF Register

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

Table 3-149. UARTF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.129 SPIA Register (Offset = 678h) [Reset = 000000C0h]

SPIA is shown in Figure 3-156 and described in Table 3-150.

Return to the Summary Table.

PER46SYSCONFIG - Peripheral System Configuration for SPIA

Figure 3-156. SPIA Register

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

Table 3-150. SPIA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.130 SPIB Register (Offset = 67Ch) [Reset = 000000C0h]

SPIB is shown in Figure 3-157 and described in Table 3-151.

Return to the Summary Table.

PER47SYSCONFIG - Peripheral System Configuration for SPIB

Figure 3-157. SPIB Register

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

Table 3-151. SPIB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.131 SPIC Register (Offset = 680h) [Reset = 000000C0h]

SPIC is shown in Figure 3-158 and described in Table 3-152.

Return to the Summary Table.

PER48SYSCONFIG - Peripheral System Configuration for SPIC

Figure 3-158. SPIC Register

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

Table 3-152. SPIC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.132 SPID Register (Offset = 684h) [Reset = 000000C0h]

SPID is shown in Figure 3-159 and described in Table 3-153.

Return to the Summary Table.

PER49SYSCONFIG - Peripheral System Configuration for SPID

Figure 3-159. SPID Register

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

Table 3-153. SPID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.133 SPIE Register (Offset = 688h) [Reset = 000000C0h]

SPIE is shown in Figure 3-160 and described in Table 3-154.

Return to the Summary Table.

PER50SYSCONFIG - Peripheral System Configuration for SPIE

Figure 3-160. SPIE Register

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

Table 3-154. SPIE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.134 I2CA Register (Offset = 68Ch) [Reset = 000000C0h]

I2CA is shown in Figure 3-161 and described in Table 3-155.

Return to the Summary Table.

PER51SYSCONFIG - Peripheral System Configuration for I2CA

Figure 3-161. I2CA Register

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

Table 3-155. I2CA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.135 I2CB Register (Offset = 690h) [Reset = 000000C0h]

I2CB is shown in Figure 3-162 and described in Table 3-156.

Return to the Summary Table.

PER52SYSCONFIG - Peripheral System Configuration for I2CB

Figure 3-162. I2CB Register

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

Table 3-156. I2CB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.136 PMBUSA Register (Offset = 694h) [Reset = 000000C0h]

PMBUSA is shown in Figure 3-163 and described in Table 3-157.

Return to the Summary Table.

PER53SYSCONFIG - Peripheral System Configuration for PMBUSA

Figure 3-163. PMBUSA Register

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

Table 3-157. PMBUSA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.137 LINA Register (Offset = 698h) [Reset = 000000C0h]

LINA is shown in Figure 3-164 and described in Table 3-158.

Return to the Summary Table.

PER54SYSCONFIG - Peripheral System Configuration for LINA

Figure 3-164. LINA Register

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

Table 3-158. LINA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.138 LINB Register (Offset = 69Ch) [Reset = 000000C0h]

LINB is shown in Figure 3-165 and described in Table 3-159.

Return to the Summary Table.

PER55SYSCONFIG - Peripheral System Configuration for LINB

Figure 3-165. LINB Register

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

Table 3-159. LINB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.139 MCANA Register (Offset = 6A0h) [Reset = 000000C0h]

MCANA is shown in Figure 3-166 and described in Table 3-160.

Return to the Summary Table.

PER56SYSCONFIG - Peripheral System Configuration for MCANA

Figure 3-166. MCANA Register

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

Table 3-160. MCANA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.140 MCANB Register (Offset = 6A4h) [Reset = 000000C0h]

MCANB is shown in Figure 3-167 and described in Table 3-161.

Return to the Summary Table.

PER57SYSCONFIG - Peripheral System Configuration for MCANB

Figure 3-167. MCANB Register

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

Table 3-161. MCANB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.141 MCANC Register (Offset = 6A8h) [Reset = 000000C0h]

MCANC is shown in Figure 3-168 and described in Table 3-162.

Return to the Summary Table.

PER58SYSCONFIG - Peripheral System Configuration for MCANC

Figure 3-168. MCANC Register

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

Table 3-162. MCANC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.142 MCAND Register (Offset = 6ACh) [Reset = 000000C0h]

MCAND is shown in Figure 3-169 and described in Table 3-163.

Return to the Summary Table.

PER59SYSCONFIG - Peripheral System Configuration for MCAND

Figure 3-169. MCAND Register

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

Table 3-163. MCAND Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.143 MCANE Register (Offset = 6B0h) [Reset = 000000C0h]

MCANE is shown in Figure 3-170 and described in Table 3-164.

Return to the Summary Table.

PER60SYSCONFIG - Peripheral System Configuration for MCANE

Figure 3-170. MCANE Register

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

Table 3-164. MCANE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.144 MCANF Register (Offset = 6B4h) [Reset = 000000C0h]

MCANF is shown in Figure 3-171 and described in Table 3-165.

Return to the Summary Table.

PER61SYSCONFIG - Peripheral System Configuration for MCANF

Figure 3-171. MCANF Register

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

Table 3-165. MCANF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.145 ADCA Register (Offset = 6B8h) [Reset = 000000C0h]

ADCA is shown in Figure 3-172 and described in Table 3-166.

Return to the Summary Table.

PER62SYSCONFIG - Peripheral System Configuration for ADCA

Figure 3-172. ADCA Register

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

Table 3-166. ADCA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.146 ADCB Register (Offset = 6BCh) [Reset = 000000C0h]

ADCB is shown in Figure 3-173 and described in Table 3-167.

Return to the Summary Table.

PER63SYSCONFIG - Peripheral System Configuration for ADCB

Figure 3-173. ADCB Register

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

Table 3-167. ADCB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.147 ADCC Register (Offset = 6C0h) [Reset = 000000C0h]

ADCC is shown in Figure 3-174 and described in Table 3-168.

Return to the Summary Table.

PER64SYSCONFIG - Peripheral System Configuration for ADCC

Figure 3-174. ADCC Register

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

Table 3-168. ADCC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.148 ADCD Register (Offset = 6C4h) [Reset = 000000C0h]

ADCD is shown in Figure 3-175 and described in Table 3-169.

Return to the Summary Table.

PER65SYSCONFIG - Peripheral System Configuration for ADCD

Figure 3-175. ADCD Register

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

Table 3-169. ADCD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.149 ADCE Register (Offset = 6C8h) [Reset = 000000C0h]

ADCE is shown in Figure 3-176 and described in Table 3-170.

Return to the Summary Table.

PER66SYSCONFIG - Peripheral System Configuration for ADCE

Figure 3-176. ADCE Register

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

Table 3-170. ADCE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.150 CMPSS1 Register (Offset = 6CCh) [Reset = 000000C0h]

CMPSS1 is shown in Figure 3-177 and described in Table 3-171.

Return to the Summary Table.

PER67SYSCONFIG - Peripheral System Configuration for CMPSS1

Figure 3-177. CMPSS1 Register

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

Table 3-171. CMPSS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.151 CMPSS2 Register (Offset = 6D0h) [Reset = 000000C0h]

CMPSS2 is shown in Figure 3-178 and described in Table 3-172.

Return to the Summary Table.

PER68SYSCONFIG - Peripheral System Configuration for CMPSS2

Figure 3-178. CMPSS2 Register

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

Table 3-172. CMPSS2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.152 CMPSS3 Register (Offset = 6D4h) [Reset = 000000C0h]

CMPSS3 is shown in Figure 3-179 and described in Table 3-173.

Return to the Summary Table.

PER69SYSCONFIG - Peripheral System Configuration for CMPSS3

Figure 3-179. CMPSS3 Register

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

Table 3-173. CMPSS3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.153 CMPSS4 Register (Offset = 6D8h) [Reset = 000000C0h]

CMPSS4 is shown in Figure 3-180 and described in Table 3-174.

Return to the Summary Table.

PER70SYSCONFIG - Peripheral System Configuration for CMPSS4

Figure 3-180. CMPSS4 Register

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

Table 3-174. CMPSS4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.154 CMPSS5 Register (Offset = 6DCh) [Reset = 000000C0h]

CMPSS5 is shown in Figure 3-181 and described in Table 3-175.

Return to the Summary Table.

PER71SYSCONFIG - Peripheral System Configuration for CMPSS5

Figure 3-181. CMPSS5 Register

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

Table 3-175. CMPSS5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.155 CMPSS6 Register (Offset = 6E0h) [Reset = 000000C0h]

CMPSS6 is shown in Figure 3-182 and described in Table 3-176.

Return to the Summary Table.

PER72SYSCONFIG - Peripheral System Configuration for CMPSS6

Figure 3-182. CMPSS6 Register

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

Table 3-176. CMPSS6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.156 CMPSS7 Register (Offset = 6E4h) [Reset = 000000C0h]

CMPSS7 is shown in Figure 3-183 and described in Table 3-177.

Return to the Summary Table.

PER73SYSCONFIG - Peripheral System Configuration for CMPSS7

Figure 3-183. CMPSS7 Register

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

Table 3-177. CMPSS7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.157 CMPSS8 Register (Offset = 6E8h) [Reset = 000000C0h]

CMPSS8 is shown in Figure 3-184 and described in Table 3-178.

Return to the Summary Table.

PER74SYSCONFIG - Peripheral System Configuration for CMPSS8

Figure 3-184. CMPSS8 Register

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

Table 3-178. CMPSS8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.158 CMPSS9 Register (Offset = 6ECh) [Reset = 000000C0h]

CMPSS9 is shown in Figure 3-185 and described in Table 3-179.

Return to the Summary Table.

PER75SYSCONFIG - Peripheral System Configuration for CMPSS9

Figure 3-185. CMPSS9 Register

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

Table 3-179. CMPSS9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.159 CMPSS10 Register (Offset = 6F0h) [Reset = 000000C0h]

CMPSS10 is shown in Figure 3-186 and described in Table 3-180.

Return to the Summary Table.

PER76SYSCONFIG - Peripheral System Configuration for CMPSS10

Figure 3-186. CMPSS10 Register

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

Table 3-180. CMPSS10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.160 CMPSS11 Register (Offset = 6F4h) [Reset = 000000C0h]

CMPSS11 is shown in Figure 3-187 and described in Table 3-181.

Return to the Summary Table.

PER77SYSCONFIG - Peripheral System Configuration for CMPSS11

Figure 3-187. CMPSS11 Register

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

Table 3-181. CMPSS11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.161 CMPSS12 Register (Offset = 6F8h) [Reset = 000000C0h]

CMPSS12 is shown in Figure 3-188 and described in Table 3-182.

Return to the Summary Table.

PER78SYSCONFIG - Peripheral System Configuration for CMPSS12

Figure 3-188. CMPSS12 Register

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

Table 3-182. CMPSS12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.162 DACA Register (Offset = 6FCh) [Reset = 000000C0h]

DACA is shown in Figure 3-189 and described in Table 3-183.

Return to the Summary Table.

PER79SYSCONFIG - Peripheral System Configuration for DACA

Figure 3-189. DACA Register

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

Table 3-183. DACA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.163 DACB Register (Offset = 700h) [Reset = 000000C0h]

DACB is shown in Figure 3-190 and described in Table 3-184.

Return to the Summary Table.

PER80SYSCONFIG - Peripheral System Configuration for DACB

Figure 3-190. DACB Register

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

Table 3-184. DACB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.164 CLB1 Register (Offset = 704h) [Reset = 000000C0h]

CLB1 is shown in Figure 3-191 and described in Table 3-185.

Return to the Summary Table.

PER81SYSCONFIG - Peripheral System Configuration for CLB1

Figure 3-191. CLB1 Register

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

Table 3-185. CLB1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.165 CLB2 Register (Offset = 708h) [Reset = 000000C0h]

CLB2 is shown in Figure 3-192 and described in Table 3-186.

Return to the Summary Table.

PER82SYSCONFIG - Peripheral System Configuration for CLB2

Figure 3-192. CLB2 Register

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

Table 3-186. CLB2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.166 CLB3 Register (Offset = 70Ch) [Reset = 000000C0h]

CLB3 is shown in Figure 3-193 and described in Table 3-187.

Return to the Summary Table.

PER83SYSCONFIG - Peripheral System Configuration for CLB3

Figure 3-193. CLB3 Register

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

Table 3-187. CLB3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.167 CLB4 Register (Offset = 710h) [Reset = 000000C0h]

CLB4 is shown in Figure 3-194 and described in Table 3-188.

Return to the Summary Table.

PER84SYSCONFIG - Peripheral System Configuration for CLB4

Figure 3-194. CLB4 Register

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

Table 3-188. CLB4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.168 CLB5 Register (Offset = 714h) [Reset = 000000C0h]

CLB5 is shown in Figure 3-195 and described in Table 3-189.

Return to the Summary Table.

PER85SYSCONFIG - Peripheral System Configuration for CLB5

Figure 3-195. CLB5 Register

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

Table 3-189. CLB5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.169 CLB6 Register (Offset = 718h) [Reset = 000000C0h]

CLB6 is shown in Figure 3-196 and described in Table 3-190.

Return to the Summary Table.

PER86SYSCONFIG - Peripheral System Configuration for CLB6

Figure 3-196. CLB6 Register

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

Table 3-190. CLB6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.170 FSITXA Register (Offset = 71Ch) [Reset = 000000C0h]

FSITXA is shown in Figure 3-197 and described in Table 3-191.

Return to the Summary Table.

PER87SYSCONFIG - Peripheral System Configuration for FSITXA

Figure 3-197. FSITXA Register

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

Table 3-191. FSITXA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.171 FSITXB Register (Offset = 720h) [Reset = 000000C0h]

FSITXB is shown in Figure 3-198 and described in Table 3-192.

Return to the Summary Table.

PER88SYSCONFIG - Peripheral System Configuration for FSITXB

Figure 3-198. FSITXB Register

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

Table 3-192. FSITXB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.172 FSITXC Register (Offset = 724h) [Reset = 000000C0h]

FSITXC is shown in Figure 3-199 and described in Table 3-193.

Return to the Summary Table.

PER89SYSCONFIG - Peripheral System Configuration for FSITXC

Figure 3-199. FSITXC Register

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

Table 3-193. FSITXC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.173 FSITXD Register (Offset = 728h) [Reset = 000000C0h]

FSITXD is shown in Figure 3-200 and described in Table 3-194.

Return to the Summary Table.

PER90SYSCONFIG - Peripheral System Configuration for FSITXD

Figure 3-200. FSITXD Register

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

Table 3-194. FSITXD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.174 FSIRXA Register (Offset = 72Ch) [Reset = 000000C0h]

FSIRXA is shown in Figure 3-201 and described in Table 3-195.

Return to the Summary Table.

PER91SYSCONFIG - Peripheral System Configuration for FSIRXA

Figure 3-201. FSIRXA Register

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

Table 3-195. FSIRXA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.175 FSIRXB Register (Offset = 730h) [Reset = 000000C0h]

FSIRXB is shown in Figure 3-202 and described in Table 3-196.

Return to the Summary Table.

PER92SYSCONFIG - Peripheral System Configuration for FSIRXB

Figure 3-202. FSIRXB Register

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

Table 3-196. FSIRXB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.176 FSIRXC Register (Offset = 734h) [Reset = 000000C0h]

FSIRXC is shown in Figure 3-203 and described in Table 3-197.

Return to the Summary Table.

PER93SYSCONFIG - Peripheral System Configuration for FSIRXC

Figure 3-203. FSIRXC Register

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

Table 3-197. FSIRXC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.177 FSIRXD Register (Offset = 738h) [Reset = 000000C0h]

FSIRXD is shown in Figure 3-204 and described in Table 3-198.

Return to the Summary Table.

PER94SYSCONFIG - Peripheral System Configuration for FSIRXD

Figure 3-204. FSIRXD Register

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

Table 3-198. FSIRXD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.178 DCC1 Register (Offset = 73Ch) [Reset = 000000C0h]

DCC1 is shown in Figure 3-205 and described in Table 3-199.

Return to the Summary Table.

PER95SYSCONFIG - Peripheral System Configuration for DCC1

Figure 3-205. DCC1 Register

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

Table 3-199. DCC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.179 DCC2 Register (Offset = 740h) [Reset = 000000C0h]

DCC2 is shown in Figure 3-206 and described in Table 3-200.

Return to the Summary Table.

PER96SYSCONFIG - Peripheral System Configuration for DCC2

Figure 3-206. DCC2 Register

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

Table 3-200. DCC2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.180 DCC3 Register (Offset = 744h) [Reset = 000000C0h]

DCC3 is shown in Figure 3-207 and described in Table 3-201.

Return to the Summary Table.

PER97SYSCONFIG - Peripheral System Configuration for DCC3

Figure 3-207. DCC3 Register

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

Table 3-201. DCC3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.181 ETHERCATA Register (Offset = 748h) [Reset = 000000C0h]

ETHERCATA is shown in Figure 3-208 and described in Table 3-202.

Return to the Summary Table.

PER98SYSCONFIG - Peripheral System Configuration for ETHERCATA

Figure 3-208. ETHERCATA Register

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

Table 3-202. ETHERCATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.182 EPG1 Register (Offset = 74Ch) [Reset = 000000C0h]

EPG1 is shown in Figure 3-209 and described in Table 3-203.

Return to the Summary Table.

PER99SYSCONFIG - Peripheral System Configuration for EPG1

Figure 3-209. EPG1 Register

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

Table 3-203. EPG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.183 SENT1 Register (Offset = 750h) [Reset = 000000C0h]

SENT1 is shown in Figure 3-210 and described in Table 3-204.

Return to the Summary Table.

PER100SYSCONFIG - Peripheral System Configuration for SENT1

Figure 3-210. SENT1 Register

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

Table 3-204. SENT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.184 SENT2 Register (Offset = 754h) [Reset = 000000C0h]

SENT2 is shown in Figure 3-211 and described in Table 3-205.

Return to the Summary Table.

PER101SYSCONFIG - Peripheral System Configuration for SENT2

Figure 3-211. SENT2 Register

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

Table 3-205. SENT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.185 SENT3 Register (Offset = 758h) [Reset = 000000C0h]

SENT3 is shown in Figure 3-212 and described in Table 3-206.

Return to the Summary Table.

PER102SYSCONFIG - Peripheral System Configuration for SENT3

Figure 3-212. SENT3 Register

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

Table 3-206. SENT3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.186 SENT4 Register (Offset = 75Ch) [Reset = 000000C0h]

SENT4 is shown in Figure 3-213 and described in Table 3-207.

Return to the Summary Table.

PER103SYSCONFIG - Peripheral System Configuration for SENT4

Figure 3-213. SENT4 Register

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

Table 3-207. SENT4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.187 SENT5 Register (Offset = 760h) [Reset = 000000C0h]

SENT5 is shown in Figure 3-214 and described in Table 3-208.

Return to the Summary Table.

PER104SYSCONFIG - Peripheral System Configuration for SENT5

Figure 3-214. SENT5 Register

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

Table 3-208. SENT5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.188 SENT6 Register (Offset = 764h) [Reset = 000000C0h]

SENT6 is shown in Figure 3-215 and described in Table 3-209.

Return to the Summary Table.

PER105SYSCONFIG - Peripheral System Configuration for SENT6

Figure 3-215. SENT6 Register

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

Table 3-209. SENT6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.189 ADCCHECKER1 Register (Offset = 768h) [Reset = 000000C0h]

ADCCHECKER1 is shown in Figure 3-216 and described in Table 3-210.

Return to the Summary Table.

PER106SYSCONFIG - Peripheral System Configuration for ADCCHECKER1

Figure 3-216. ADCCHECKER1 Register

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

Table 3-210. ADCCHECKER1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.190 ADCCHECKER2 Register (Offset = 76Ch) [Reset = 000000C0h]

ADCCHECKER2 is shown in Figure 3-217 and described in Table 3-211.

Return to the Summary Table.

PER107SYSCONFIG - Peripheral System Configuration for ADCCHECKER2

Figure 3-217. ADCCHECKER2 Register

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

Table 3-211. ADCCHECKER2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.191 ADCCHECKER3 Register (Offset = 770h) [Reset = 000000C0h]

ADCCHECKER3 is shown in Figure 3-218 and described in Table 3-212.

Return to the Summary Table.

PER108SYSCONFIG - Peripheral System Configuration for ADCCHECKER3

Figure 3-218. ADCCHECKER3 Register

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

Table 3-212. ADCCHECKER3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.192 ADCCHECKER4 Register (Offset = 774h) [Reset = 000000C0h]

ADCCHECKER4 is shown in Figure 3-219 and described in Table 3-213.

Return to the Summary Table.

PER109SYSCONFIG - Peripheral System Configuration for ADCCHECKER4

Figure 3-219. ADCCHECKER4 Register

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

Table 3-213. ADCCHECKER4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.193 ADCCHECKER5 Register (Offset = 778h) [Reset = 000000C0h]

ADCCHECKER5 is shown in Figure 3-220 and described in Table 3-214.

Return to the Summary Table.

PER110SYSCONFIG - Peripheral System Configuration for ADCCHECKER5

Figure 3-220. ADCCHECKER5 Register

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

Table 3-214. ADCCHECKER5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.194 ADCCHECKER6 Register (Offset = 77Ch) [Reset = 000000C0h]

ADCCHECKER6 is shown in Figure 3-221 and described in Table 3-215.

Return to the Summary Table.

PER111SYSCONFIG - Peripheral System Configuration for ADCCHECKER6

Figure 3-221. ADCCHECKER6 Register

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

Table 3-215. ADCCHECKER6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.195 ADCCHECKER7 Register (Offset = 780h) [Reset = 000000C0h]

ADCCHECKER7 is shown in Figure 3-222 and described in Table 3-216.

Return to the Summary Table.

PER112SYSCONFIG - Peripheral System Configuration for ADCCHECKER7

Figure 3-222. ADCCHECKER7 Register

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

Table 3-216. ADCCHECKER7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.196 ADCCHECKER8 Register (Offset = 784h) [Reset = 000000C0h]

ADCCHECKER8 is shown in Figure 3-223 and described in Table 3-217.

Return to the Summary Table.

PER113SYSCONFIG - Peripheral System Configuration for ADCCHECKER8

Figure 3-223. ADCCHECKER8 Register

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

Table 3-217. ADCCHECKER8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.197 ADCCHECKER9 Register (Offset = 788h) [Reset = 000000C0h]

ADCCHECKER9 is shown in Figure 3-224 and described in Table 3-218.

Return to the Summary Table.

PER114SYSCONFIG - Peripheral System Configuration for ADCCHECKER9

Figure 3-224. ADCCHECKER9 Register

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

Table 3-218. ADCCHECKER9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.198 ADCCHECKER10 Register (Offset = 78Ch) [Reset = 000000C0h]

ADCCHECKER10 is shown in Figure 3-225 and described in Table 3-219.

Return to the Summary Table.

PER115SYSCONFIG - Peripheral System Configuration for ADCCHECKER10

Figure 3-225. ADCCHECKER10 Register

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

Table 3-219. ADCCHECKER10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.199 ADCSEAGGRCPU1 Register (Offset = 790h) [Reset = 000000C0h]

ADCSEAGGRCPU1 is shown in Figure 3-226 and described in Table 3-220.

Return to the Summary Table.

PER116SYSCONFIG - Peripheral System Configuration for ADCSEAGGRCPU1

Figure 3-226. ADCSEAGGRCPU1 Register

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

Table 3-220. ADCSEAGGRCPU1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.200 ADCSEAGGRCPU2 Register (Offset = 794h) [Reset = 000000C0h]

ADCSEAGGRCPU2 is shown in Figure 3-227 and described in Table 3-221.

Return to the Summary Table.

PER117SYSCONFIG - Peripheral System Configuration for ADCSEAGGRCPU2

Figure 3-227. ADCSEAGGRCPU2 Register

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

Table 3-221. ADCSEAGGRCPU2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.201 ADCSEAGGRCPU3 Register (Offset = 798h) [Reset = 000000C0h]

ADCSEAGGRCPU3 is shown in Figure 3-228 and described in Table 3-222.

Return to the Summary Table.

PER118SYSCONFIG - Peripheral System Configuration for ADCSEAGGRCPU3

Figure 3-228. ADCSEAGGRCPU3 Register

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

Table 3-222. ADCSEAGGRCPU3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.202 RTDMA1CH Register (Offset = 7A8h) [Reset = 000000C0h]

RTDMA1CH is shown in Figure 3-229 and described in Table 3-223.

Return to the Summary Table.

PER122SYSCONFIG - Peripheral System Configuration for RTDMA1CH

Figure 3-229. RTDMA1CH Register

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

Table 3-223. RTDMA1CH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.203 RTDMA2CH Register (Offset = 7ACh) [Reset = 000000C0h]

RTDMA2CH is shown in Figure 3-230 and described in Table 3-224.

Return to the Summary Table.

PER123SYSCONFIG - Peripheral System Configuration for RTDMA2CH

Figure 3-230. RTDMA2CH Register

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

Table 3-224. RTDMA2CH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.204 WADI1 Register (Offset = 7B0h) [Reset = 000000C0h]

WADI1 is shown in Figure 3-231 and described in Table 3-225.

Return to the Summary Table.

PER124SYSCONFIG - Peripheral System Configuration for WADI1

Figure 3-231. WADI1 Register

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

Table 3-225. WADI1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.205 WADI2 Register (Offset = 7B4h) [Reset = 000000C0h]

WADI2 is shown in Figure 3-232 and described in Table 3-226.

Return to the Summary Table.

PER125SYSCONFIG - Peripheral System Configuration for WADI2

Figure 3-232. WADI2 Register

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

Table 3-226. WADI2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.206 INPUTXBARFlags Register (Offset = 7B8h) [Reset = 000000C0h]

INPUTXBARFlags is shown in Figure 3-233 and described in Table 3-227.

Return to the Summary Table.

PER126SYSCONFIG - Peripheral System Configuration for INPUTXBARFlags

Figure 3-233. INPUTXBARFlags Register

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

Table 3-227. INPUTXBARFlags Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.207 OUTPUTXBARFlags Register (Offset = 7BCh) [Reset = 000000C0h]

OUTPUTXBARFlags is shown in Figure 3-234 and described in Table 3-228.

Return to the Summary Table.

PER127SYSCONFIG - Peripheral System Configuration for OUTPUTXBARFlags

Figure 3-234. OUTPUTXBARFlags Register

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

Table 3-228. OUTPUTXBARFlags Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.208 DLTFIFORegs Register (Offset = 7C0h) [Reset = 000000C0h]

DLTFIFORegs is shown in Figure 3-235 and described in Table 3-229.

Return to the Summary Table.

PER128SYSCONFIG - Peripheral System Configuration for DLTFIFORegs

Figure 3-235. DLTFIFORegs Register

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

Table 3-229. DLTFIFORegs Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.209 ADC_GLOBAL_REGS Register (Offset = 7C4h) [Reset = 000000C0h]

ADC_GLOBAL_REGS is shown in Figure 3-236 and described in Table 3-230.

Return to the Summary Table.

PER129SYSCONFIG - Peripheral System Configuration for ADC_GLOBAL_REGS

Figure 3-236. ADC_GLOBAL_REGS Register

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

Table 3-230. ADC_GLOBAL_REGS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.210 Error_Aggregator Register (Offset = 7C8h) [Reset = 000000C0h]

Error_Aggregator is shown in Figure 3-237 and described in Table 3-231.

Return to the Summary Table.

PER130SYSCONFIG - Peripheral System Configuration for Error_Aggregator

Figure 3-237. Error_Aggregator Register

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

Table 3-231. Error_Aggregator Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.211 ESM Register (Offset = 7CCh) [Reset = 000000C0h]

ESM is shown in Figure 3-238 and described in Table 3-232.

Return to the Summary Table.

PER131SYSCONFIG - Peripheral System Configuration for ESM ESMCPU1/2/3 and ESMSYS

Figure 3-238. ESM Register

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

Table 3-232. ESM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	DBGHALTEN	R/W	1h	0 : Peripheral DBGHALT signal NOT activated when respective CPU (selected by CPUSEL) enters HALT mode 1 : Peripheral DBGHALT signal activated when respective CPU (selected by CPUSEL) enters HALT mode Reset type: CPU1.SYSRSn
6	STANDBYEN	R/W	1h	0 : Peripheral NOT clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode 1 : Peripheral clock gated when respective CPU (selected by CPUSEL) enters STANDBY mode Reset type: CPU1.SYSRSn
5-3	CPUSEL	R/W	0h	Peripheral CPU selection logic 000: Connected to CPU1 (default) 001: Connected to CPU2 010: Connected to CPU3 011: Reserved 1xx: Reserved Reset type: CPU1.SYSRSn
2-0	FRAMESEL	R/W	0h	Peripheral selection logic for FRAME 000: Connected to FRAME0 (default) 001: Connected to FRAME1 010: Connected to FRAME2 011: Connected to FRAME3 1xx: Reserved Reset type: CPU1.SYSRSn

3.13.2.212 PARITY_TEST Register (Offset = 7E4h) [Reset = 00000000h]

PARITY_TEST is shown in Figure 3-239 and described in Table 3-233.

Return to the Summary Table.

Enables parity test

Figure 3-239. PARITY_TEST Register

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

Table 3-233. PARITY_TEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	TESTEN	R/W	0h	1010: Parity test feature is enabled All other values: Parity test feature is disabled Note: (1) When the parity test feature is enabled, actual registers are not accessible in the memory map. Instead, the parity values (final XOR output indicating the parity error) are accessible. Parity is computed for every byte and the corresponding parity error value is available at the bit-0 of every byte. Value of '1' written to the parity bit after enabling the parity test feature can be used to inject the error by inverting the stored parity value. (2) It is recommended to leave the field as 0101 or 0000 after completing the parity test. Reset type: SYSRSn

3.13.3 MEMSS_L_CONFIG_REGS Registers

Table 3-234 lists the memory-mapped registers for the MEMSS_L_CONFIG_REGS registers. All register offset addresses not listed in Table 3-234 should be considered as reserved locations and the register contents should not be modified.

Table 3-234. MEMSS_L_CONFIG_REGS Registers

Offset	Acronym	Register Name	Protection
0h	LPA0_MEM_CONFIG	LPA0 Memory Configuration Register	PARITY
4h	LPA0_MEM_CONFIG_LOCK	Temporary Lock for LPA0 Memory Configuration Register	PARITY
8h	LPA0_MEM_CONFIG_COMMIT	Permament Lock for LPA0 Memory Configuration Register	PARITY
10h	LPA1_MEM_CONFIG	LPA1 Memory Configuration Register	PARITY
14h	LPA1_MEM_CONFIG_LOCK	Temporary Lock for LPA1 Memory Configuration Register	PARITY
18h	LPA1_MEM_CONFIG_COMMIT	Permament Lock for LPA1 Memory Configuration Register	PARITY
20h	LDA0_MEM_CONFIG	LDA0 Memory Configuration Register	PARITY
24h	LDA0_MEM_CONFIG_LOCK	Temporary Lock for LDA0 Memory Configuration Register	PARITY
28h	LDA0_MEM_CONFIG_COMMIT	Permament Lock for LDA0 Memory Configuration Register	PARITY
30h	LDA1_MEM_CONFIG	LDA1 Memory Configuration Register	PARITY
34h	LDA1_MEM_CONFIG_LOCK	Temporary Lock for LDA1 Memory Configuration Register	PARITY
38h	LDA1_MEM_CONFIG_COMMIT	Permament Lock for LDA1 Memory Configuration Register	PARITY
40h	LDA2_MEM_CONFIG	LDA2 Memory Configuration Register	PARITY
44h	LDA2_MEM_CONFIG_LOCK	Temporary Lock for LDA2 Memory Configuration Register	PARITY
48h	LDA2_MEM_CONFIG_COMMIT	Permament Lock for LDA2 Memory Configuration Register	PARITY
50h	LDA3_MEM_CONFIG	LDA3 Memory Configuration Register	PARITY
54h	LDA3_MEM_CONFIG_LOCK	Temporary Lock for LDA3 Memory Configuration Register	PARITY
58h	LDA3_MEM_CONFIG_COMMIT	Permament Lock for LDA3 Memory Configuration Register	PARITY
60h	LDA4_MEM_CONFIG	LDA4 Memory Configuration Register	PARITY
64h	LDA4_MEM_CONFIG_LOCK	Temporary Lock for LDA4 Memory Configuration Register	PARITY
68h	LDA4_MEM_CONFIG_COMMIT	Permament Lock for LDA4 Memory Configuration Register	PARITY
70h	LDA5_MEM_CONFIG	LDA5 Memory Configuration Register	PARITY
74h	LDA5_MEM_CONFIG_LOCK	Temporary Lock for LDA5 Memory Configuration Register	PARITY
78h	LDA5_MEM_CONFIG_COMMIT	Permament Lock for LDA5 Memory Configuration Register	PARITY
80h	LDA6_MEM_CONFIG	LDA6 Memory Configuration Register	PARITY
84h	LDA6_MEM_CONFIG_LOCK	Temporary Lock for LDA6 Memory Configuration Register	PARITY
88h	LDA6_MEM_CONFIG_COMMIT	Permament Lock for LDA6 Memory Configuration Register	PARITY
90h	LDA7_MEM_CONFIG	LDA7 Memory Configuration Register	PARITY
94h	LDA7_MEM_CONFIG_LOCK	Temporary Lock for LDA7 Memory Configuration Register	PARITY
98h	LDA7_MEM_CONFIG_COMMIT	Permament Lock for LDA7 Memory Configuration Register	PARITY

Complex bit access types are encoded to fit into small table cells. Table 3-235 shows the codes that are used for access types in this section.

Table 3-235. MEMSS_L_CONFIG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value

3.13.3.1 LPA0_MEM_CONFIG Register (Offset = 0h) [Reset = 00000000h]

LPA0_MEM_CONFIG is shown in Figure 3-240 and described in Table 3-236.

Return to the Summary Table.

LPA0 Memory Configuration Register

Figure 3-240. LPA0_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-236. LPA0_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : LPA0 memory is not Intialized 1 : LPA0 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.3.2 LPA0_MEM_CONFIG_LOCK Register (Offset = 4h) [Reset = 00000000h]

LPA0_MEM_CONFIG_LOCK is shown in Figure 3-241 and described in Table 3-237.

Return to the Summary Table.

Temporary Lock for LPA0 Memory Configuration Register

Figure 3-241. LPA0_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LPA0_MEM_CONFIG
R-0h							R/W-0h

Table 3-237. LPA0_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LPA0_MEM_CONFIG	R/W	0h	0 : Write to LPA0_MEM_CONFIG is allowed. 1 : Write to LPA0_MEM_CONFIG is not allowed. Note : This bit can only be modified if LPA0_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.3.3 LPA0_MEM_CONFIG_COMMIT Register (Offset = 8h) [Reset = 00000000h]

LPA0_MEM_CONFIG_COMMIT is shown in Figure 3-242 and described in Table 3-238.

Return to the Summary Table.

Permament Lock for LPA0 Memory Configuration Register

Figure 3-242. LPA0_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LPA0_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-238. LPA0_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LPA0_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the LPA0_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset 0 : LPA0_MEM_CONFIG is modifiable 1 : LPA0_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.3.4 LPA1_MEM_CONFIG Register (Offset = 10h) [Reset = 00000000h]

LPA1_MEM_CONFIG is shown in Figure 3-243 and described in Table 3-239.

Return to the Summary Table.

LPA1 Memory Configuration Register

Figure 3-243. LPA1_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-239. LPA1_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : LPA1 memory is not Intialized 1 : LPA1 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.3.5 LPA1_MEM_CONFIG_LOCK Register (Offset = 14h) [Reset = 00000000h]

LPA1_MEM_CONFIG_LOCK is shown in Figure 3-244 and described in Table 3-240.

Return to the Summary Table.

Temporary Lock for LPA1 Memory Configuration Register

Figure 3-244. LPA1_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LPA1_MEM_CONFIG
R-0h							R/W-0h

Table 3-240. LPA1_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LPA1_MEM_CONFIG	R/W	0h	0 : Write to LPA1_MEM_CONFIG is allowed. 1 : Write to LPA1_MEM_CONFIG is not allowed. Note : This bit can only be modified if LPA1_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.3.6 LPA1_MEM_CONFIG_COMMIT Register (Offset = 18h) [Reset = 00000000h]

LPA1_MEM_CONFIG_COMMIT is shown in Figure 3-245 and described in Table 3-241.

Return to the Summary Table.

Permament Lock for LPA1 Memory Configuration Register

Figure 3-245. LPA1_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LPA1_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-241. LPA1_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LPA1_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the LPA1_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset 0 : LPA1_MEM_CONFIG is modifiable 1 : LPA1_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.3.7 LDA0_MEM_CONFIG Register (Offset = 20h) [Reset = 00000000h]

LDA0_MEM_CONFIG is shown in Figure 3-246 and described in Table 3-242.

Return to the Summary Table.

LDA0 Memory Configuration Register

Figure 3-246. LDA0_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						TESTMODE	
R-0h						R/W-0h	

Table 3-242. LDA0_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : LDA0 memory is not Intialized 1 : LDA0 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.3.8 LDA0_MEM_CONFIG_LOCK Register (Offset = 24h) [Reset = 00000000h]

LDA0_MEM_CONFIG_LOCK is shown in Figure 3-247 and described in Table 3-243.

Return to the Summary Table.

Temporary Lock for LDA0 Memory Configuration Register

Figure 3-247. LDA0_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LDA0_MEM_CONFIG
R-0h							R/W-0h

Table 3-243. LDA0_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LDA0_MEM_CONFIG	R/W	0h	0 : Write to LDA0_MEM_CONFIG is allowed. 1 : Write to LDA0_MEM_CONFIG is not allowed. Note : This bit can only be modified if LDA0_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.3.9 LDA0_MEM_CONFIG_COMMIT Register (Offset = 28h) [Reset = 00000000h]

LDA0_MEM_CONFIG_COMMIT is shown in Figure 3-248 and described in Table 3-244.

Return to the Summary Table.

Permament Lock for LDA0 Memory Configuration Register

Figure 3-248. LDA0_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LDA0_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-244. LDA0_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LDA0_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the LDA0_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset 0 : LDA0_MEM_CONFIG is modifiable 1 : LDA0_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.3.10 LDA1_MEM_CONFIG Register (Offset = 30h) [Reset = 00000000h]

LDA1_MEM_CONFIG is shown in Figure 3-249 and described in Table 3-245.

Return to the Summary Table.

LDA1 Memory Configuration Register

Figure 3-249. LDA1_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-245. LDA1_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : LDA1 memory is not Intialized 1 : LDA1 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.3.11 LDA1_MEM_CONFIG_LOCK Register (Offset = 34h) [Reset = 00000000h]

LDA1_MEM_CONFIG_LOCK is shown in Figure 3-250 and described in Table 3-246.

Return to the Summary Table.

Temporary Lock for LDA1 Memory Configuration Register

Figure 3-250. LDA1_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LDA1_MEM_CONFIG
R-0h							R/W-0h

Table 3-246. LDA1_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LDA1_MEM_CONFIG	R/W	0h	0 : Write to LDA1_MEM_CONFIG is allowed. 1 : Write to LDA1_MEM_CONFIG is not allowed. Note : This bit can only be modified if LDA1_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.3.12 LDA1_MEM_CONFIG_COMMIT Register (Offset = 38h) [Reset = 00000000h]

LDA1_MEM_CONFIG_COMMIT is shown in Figure 3-251 and described in Table 3-247.

Return to the Summary Table.

Permament Lock for LDA1 Memory Configuration Register

Figure 3-251. LDA1_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LDA1_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-247. LDA1_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LDA1_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the LDA1_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset 0 : LDA1_MEM_CONFIG is modifiable 1 : LDA1_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.3.13 LDA2_MEM_CONFIG Register (Offset = 40h) [Reset = 00000000h]

LDA2_MEM_CONFIG is shown in Figure 3-252 and described in Table 3-248.

Return to the Summary Table.

LDA2 Memory Configuration Register

Figure 3-252. LDA2_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-248. LDA2_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : LDA2 memory is not Intialized 1 : LDA2 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.3.14 LDA2_MEM_CONFIG_LOCK Register (Offset = 44h) [Reset = 00000000h]

LDA2_MEM_CONFIG_LOCK is shown in Figure 3-253 and described in Table 3-249.

Return to the Summary Table.

Temporary Lock for LDA2 Memory Configuration Register

Figure 3-253. LDA2_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LDA2_MEM_CONFIG
R-0h							R/W-0h

Table 3-249. LDA2_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LDA2_MEM_CONFIG	R/W	0h	0 : Write to LDA2_MEM_CONFIG is allowed. 1 : Write to LDA2_MEM_CONFIG is not allowed. Note : This bit can only be modified if LDA2_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.3.15 LDA2_MEM_CONFIG_COMMIT Register (Offset = 48h) [Reset = 00000000h]

LDA2_MEM_CONFIG_COMMIT is shown in Figure 3-254 and described in Table 3-250.

Return to the Summary Table.

Permament Lock for LDA2 Memory Configuration Register

Figure 3-254. LDA2_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LDA2_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-250. LDA2_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LDA2_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the LDA2_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset 0 : LDA2_MEM_CONFIG is modifiable 1 : LDA2_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.3.16 LDA3_MEM_CONFIG Register (Offset = 50h) [Reset = 00000000h]

LDA3_MEM_CONFIG is shown in Figure 3-255 and described in Table 3-251.

Return to the Summary Table.

LDA3 Memory Configuration Register

Figure 3-255. LDA3_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-251. LDA3_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : LDA3 memory is not Intialized 1 : LDA3 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.3.17 LDA3_MEM_CONFIG_LOCK Register (Offset = 54h) [Reset = 00000000h]

LDA3_MEM_CONFIG_LOCK is shown in Figure 3-256 and described in Table 3-252.

Return to the Summary Table.

Temporary Lock for LDA3 Memory Configuration Register

Figure 3-256. LDA3_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LDA3_MEM_CONFIG
R-0h							R/W-0h

Table 3-252. LDA3_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LDA3_MEM_CONFIG	R/W	0h	0 : Write to LDA3_MEM_CONFIG is allowed. 1 : Write to LDA3_MEM_CONFIG is not allowed. Note : This bit can only be modified if LDA3_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.3.18 LDA3_MEM_CONFIG_COMMIT Register (Offset = 58h) [Reset = 00000000h]

LDA3_MEM_CONFIG_COMMIT is shown in Figure 3-257 and described in Table 3-253.

Return to the Summary Table.

Permament Lock for LDA3 Memory Configuration Register

Figure 3-257. LDA3_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LDA3_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-253. LDA3_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LDA3_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the LDA3_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset 0 : LDA3_MEM_CONFIG is modifiable 1 : LDA3_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.3.19 LDA4_MEM_CONFIG Register (Offset = 60h) [Reset = 00000000h]

LDA4_MEM_CONFIG is shown in Figure 3-258 and described in Table 3-254.

Return to the Summary Table.

LDA4 Memory Configuration Register

Figure 3-258. LDA4_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-254. LDA4_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : LDA4 memory is not Intialized 1 : LDA4 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.3.20 LDA4_MEM_CONFIG_LOCK Register (Offset = 64h) [Reset = 00000000h]

LDA4_MEM_CONFIG_LOCK is shown in Figure 3-259 and described in Table 3-255.

Return to the Summary Table.

Temporary Lock for LDA4 Memory Configuration Register

Figure 3-259. LDA4_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LDA4_MEM_CONFIG
R-0h							R/W-0h

Table 3-255. LDA4_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LDA4_MEM_CONFIG	R/W	0h	0 : Write to LDA4_MEM_CONFIG is allowed. 1 : Write to LDA4_MEM_CONFIG is not allowed. Note : This bit can only be modified if LDA4_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.3.21 LDA4_MEM_CONFIG_COMMIT Register (Offset = 68h) [Reset = 00000000h]

LDA4_MEM_CONFIG_COMMIT is shown in Figure 3-260 and described in Table 3-256.

Return to the Summary Table.

Permament Lock for LDA4 Memory Configuration Register

Figure 3-260. LDA4_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LDA4_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-256. LDA4_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LDA4_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the LDA4_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset 0 : LDA4_MEM_CONFIG is modifiable 1 : LDA4_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.3.22 LDA5_MEM_CONFIG Register (Offset = 70h) [Reset = 00000000h]

LDA5_MEM_CONFIG is shown in Figure 3-261 and described in Table 3-257.

Return to the Summary Table.

LDA5 Memory Configuration Register

Figure 3-261. LDA5_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-257. LDA5_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : LDA5 memory is not Intialized 1 : LDA5 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.3.23 LDA5_MEM_CONFIG_LOCK Register (Offset = 74h) [Reset = 00000000h]

LDA5_MEM_CONFIG_LOCK is shown in Figure 3-262 and described in Table 3-258.

Return to the Summary Table.

Temporary Lock for LDA5 Memory Configuration Register

Figure 3-262. LDA5_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LDA5_MEM_CONFIG
R-0h							R/W-0h

Table 3-258. LDA5_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LDA5_MEM_CONFIG	R/W	0h	0 : Write to LDA5_MEM_CONFIG is allowed. 1 : Write to LDA5_MEM_CONFIG is not allowed. Note : This bit can only be modified if LDA5_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.3.24 LDA5_MEM_CONFIG_COMMIT Register (Offset = 78h) [Reset = 00000000h]

LDA5_MEM_CONFIG_COMMIT is shown in Figure 3-263 and described in Table 3-259.

Return to the Summary Table.

Permament Lock for LDA5 Memory Configuration Register

Figure 3-263. LDA5_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LDA5_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-259. LDA5_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LDA5_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the LDA5_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset 0 : LDA5_MEM_CONFIG is modifiable 1 : LDA5_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.3.25 LDA6_MEM_CONFIG Register (Offset = 80h) [Reset = 00000000h]

LDA6_MEM_CONFIG is shown in Figure 3-264 and described in Table 3-260.

Return to the Summary Table.

LDA6 Memory Configuration Register

Figure 3-264. LDA6_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-260. LDA6_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : LDA6 memory is not Intialized 1 : LDA6 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.3.26 LDA6_MEM_CONFIG_LOCK Register (Offset = 84h) [Reset = 00000000h]

LDA6_MEM_CONFIG_LOCK is shown in Figure 3-265 and described in Table 3-261.

Return to the Summary Table.

Temporary Lock for LDA6 Memory Configuration Register

Figure 3-265. LDA6_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LDA6_MEM_CONFIG
R-0h							R/W-0h

Table 3-261. LDA6_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LDA6_MEM_CONFIG	R/W	0h	0 : Write to LDA6_MEM_CONFIG is allowed. 1 : Write to LDA6_MEM_CONFIG is not allowed. Note : This bit can only be modified if LDA6_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.3.27 LDA6_MEM_CONFIG_COMMIT Register (Offset = 88h) [Reset = 00000000h]

LDA6_MEM_CONFIG_COMMIT is shown in Figure 3-266 and described in Table 3-262.

Return to the Summary Table.

Permament Lock for LDA6 Memory Configuration Register

Figure 3-266. LDA6_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LDA6_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-262. LDA6_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LDA6_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the LDA6_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset 0 : LDA6_MEM_CONFIG is modifiable 1 : LDA6_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.3.28 LDA7_MEM_CONFIG Register (Offset = 90h) [Reset = 00000000h]

LDA7_MEM_CONFIG is shown in Figure 3-267 and described in Table 3-263.

Return to the Summary Table.

LDA7 Memory Configuration Register

Figure 3-267. LDA7_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-263. LDA7_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : LDA7 memory is not Intialized 1 : LDA7 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.3.29 LDA7_MEM_CONFIG_LOCK Register (Offset = 94h) [Reset = 00000000h]

LDA7_MEM_CONFIG_LOCK is shown in Figure 3-268 and described in Table 3-264.

Return to the Summary Table.

Temporary Lock for LDA7 Memory Configuration Register

Figure 3-268. LDA7_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LDA7_MEM_CONFIG
R-0h							R/W-0h

Table 3-264. LDA7_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LDA7_MEM_CONFIG	R/W	0h	0 : Write to LDA7_MEM_CONFIG is allowed. 1 : Write to LDA7_MEM_CONFIG is not allowed. Note : This bit can only be modified if LDA7_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.3.30 LDA7_MEM_CONFIG_COMMIT Register (Offset = 98h) [Reset = 00000000h]

LDA7_MEM_CONFIG_COMMIT is shown in Figure 3-269 and described in Table 3-265.

Return to the Summary Table.

Permament Lock for LDA7 Memory Configuration Register

Figure 3-269. LDA7_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							LDA7_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-265. LDA7_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LDA7_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the LDA7_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset 0 : LDA7_MEM_CONFIG is modifiable 1 : LDA7_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.4 MEMSS_C_CONFIG_REGS Registers

Table 3-266 lists the memory-mapped registers for the MEMSS_C_CONFIG_REGS registers. All register offset addresses not listed in Table 3-266 should be considered as reserved locations and the register contents should not be modified.

Table 3-266. MEMSS_C_CONFIG_REGS Registers

Offset	Acronym	Register Name	Protection
0h	CPA0_MEM_CONFIG	CPA0 Memory Configuration Register	PARITY
4h	CPA0_MEM_CONFIG_LOCK	Temporary Lock for CPA0 Memory Configuration Register	PARITY
8h	CPA0_MEM_CONFIG_COMMIT	Permament Lock for CPA0 Memory Configuration Register	PARITY
10h	CPA1_MEM_CONFIG	CPA1 Memory Configuration Register	PARITY
14h	CPA1_MEM_CONFIG_LOCK	Temporary Lock for CPA1 Memory Configuration Register	PARITY
18h	CPA1_MEM_CONFIG_COMMIT	Permament Lock for CPA1 Memory Configuration Register	PARITY
20h	CDA0_MEM_CONFIG	CDA0 Memory Configuration Register	PARITY
24h	CDA0_MEM_CONFIG_LOCK	Temporary Lock for CDA0 Memory Configuration Register	PARITY
28h	CDA0_MEM_CONFIG_COMMIT	Permament Lock for CDA0 Memory Configuration Register	PARITY
30h	CDA1_MEM_CONFIG	CDA1 Memory Configuration Register	PARITY
34h	CDA1_MEM_CONFIG_LOCK	Temporary Lock for CDA1 Memory Configuration Register	PARITY
38h	CDA1_MEM_CONFIG_COMMIT	Permament Lock for CDA1 Memory Configuration Register	PARITY
40h	CDA2_MEM_CONFIG	CDA2 Memory Configuration Register	PARITY
44h	CDA2_MEM_CONFIG_LOCK	Temporary Lock for CDA2 Memory Configuration Register	PARITY
48h	CDA2_MEM_CONFIG_COMMIT	Permament Lock for CDA2 Memory Configuration Register	PARITY
50h	CDA3_MEM_CONFIG	CDA3 Memory Configuration Register	PARITY
54h	CDA3_MEM_CONFIG_LOCK	Temporary Lock for CDA3 Memory Configuration Register	PARITY
58h	CDA3_MEM_CONFIG_COMMIT	Permament Lock for CDA3 Memory Configuration Register	PARITY
60h	CDA4_MEM_CONFIG	CDA4 Memory Configuration Register	PARITY
64h	CDA4_MEM_CONFIG_LOCK	Temporary Lock for CDA4 Memory Configuration Register	PARITY
68h	CDA4_MEM_CONFIG_COMMIT	Permament Lock for CDA4 Memory Configuration Register	PARITY
70h	CDA5_MEM_CONFIG	CDA5 Memory Configuration Register	PARITY
74h	CDA5_MEM_CONFIG_LOCK	Temporary Lock for CDA5 Memory Configuration Register	PARITY
78h	CDA5_MEM_CONFIG_COMMIT	Permament Lock for CDA5 Memory Configuration Register	PARITY
80h	CDA6_MEM_CONFIG	CDA6 Memory Configuration Register	PARITY
84h	CDA6_MEM_CONFIG_LOCK	Temporary Lock for CDA6 Memory Configuration Register	PARITY
88h	CDA6_MEM_CONFIG_COMMIT	Permament Lock for CDA6 Memory Configuration Register	PARITY
90h	CDA7_MEM_CONFIG	CDA7 Memory Configuration Register	PARITY
94h	CDA7_MEM_CONFIG_LOCK	Temporary Lock for CDA7 Memory Configuration Register	PARITY
98h	CDA7_MEM_CONFIG_COMMIT	Permament Lock for CDA7 Memory Configuration Register	PARITY
A0h	CDA8_MEM_CONFIG	CDA8 Memory Configuration Register	PARITY
A4h	CDA8_MEM_CONFIG_LOCK	Temporary Lock for CDA8 Memory Configuration Register	PARITY
A8h	CDA8_MEM_CONFIG_COMMIT	Permament Lock for CDA8 Memory Configuration Register	PARITY

Table 3-266. MEMSS_C_CONFIG_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
B0h	CDA9_MEM_CONFIG	CDA9 Memory Configuration Register	PARITY
B4h	CDA9_MEM_CONFIG_LOCK	Temporary Lock for CDA9 Memory Configuration Register	PARITY
B8h	CDA9_MEM_CONFIG_COMMIT	Permament Lock for CDA9 Memory Configuration Register	PARITY
C0h	CDA10_MEM_CONFIG	CDA10 Memory Configuration Register	PARITY
C4h	CDA10_MEM_CONFIG_LOCK	Temporary Lock for CDA10 Memory Configuration Register	PARITY
C8h	CDA10_MEM_CONFIG_COMMIT	Permament Lock for CDA10 Memory Configuration Register	PARITY
D0h	CDA11_MEM_CONFIG	CDA11 Memory Configuration Register	PARITY
D4h	CDA11_MEM_CONFIG_LOCK	Temporary Lock for CDA11 Memory Configuration Register	PARITY
D8h	CDA11_MEM_CONFIG_COMMIT	Permament Lock for CDA11 Memory Configuration Register	PARITY

Complex bit access types are encoded to fit into small table cells. Table 3-267 shows the codes that are used for access types in this section.

Table 3-267. MEMSS_C_CONFIG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value

3.13.4.1 CPA0_MEM_CONFIG Register (Offset = 0h) [Reset = 00000000h]

CPA0_MEM_CONFIG is shown in Figure 3-270 and described in Table 3-268.

Return to the Summary Table.

CPA0 Memory Configuration Register

Figure 3-270. CPA0_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-268. CPA0_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : CPA0 memory is not Intialized 1 : CPA0 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.4.2 CPA0_MEM_CONFIG_LOCK Register (Offset = 4h) [Reset = 00000000h]

CPA0_MEM_CONFIG_LOCK is shown in Figure 3-271 and described in Table 3-269.

Return to the Summary Table.

Temporary Lock for CPA0 Memory Configuration Register

Figure 3-271. CPA0_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CPA0_MEM_CONFIG
R-0h							R/W-0h

Table 3-269. CPA0_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CPA0_MEM_CONFIG	R/W	0h	0 : Write to CPA0_MEM_CONFIG is allowed. 1 : Write to CPA0_MEM_CONFIG is not allowed. Note : This bit can only be modified if CPA0_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.4.3 CPA0_MEM_CONFIG_COMMIT Register (Offset = 8h) [Reset = 00000000h]

CPA0_MEM_CONFIG_COMMIT is shown in Figure 3-272 and described in Table 3-270.

Return to the Summary Table.

Permament Lock for CPA0 Memory Configuration Register

Figure 3-272. CPA0_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CPA0_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-270. CPA0_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CPA0_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the CPA0_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset. 0 : CPA0_MEM_CONFIG is modifiable 1 : CPA0_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.4.4 CPA1_MEM_CONFIG Register (Offset = 10h) [Reset = 00000000h]

CPA1_MEM_CONFIG is shown in Figure 3-273 and described in Table 3-271.

Return to the Summary Table.

CPA1 Memory Configuration Register

Figure 3-273. CPA1_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-271. CPA1_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : CPA1 memory is not Intialized 1 : CPA1 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.4.5 CPA1_MEM_CONFIG_LOCK Register (Offset = 14h) [Reset = 00000000h]

CPA1_MEM_CONFIG_LOCK is shown in Figure 3-274 and described in Table 3-272.

Return to the Summary Table.

Temporary Lock for CPA1 Memory Configuration Register

Figure 3-274. CPA1_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CPA1_MEM_CONFIG
R-0h							R/W-0h

Table 3-272. CPA1_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CPA1_MEM_CONFIG	R/W	0h	0 : Write to CPA1_MEM_CONFIG is allowed. 1 : Write to CPA1_MEM_CONFIG is not allowed. Note : This bit can only be modified if CPA1_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.4.6 CPA1_MEM_CONFIG_COMMIT Register (Offset = 18h) [Reset = 00000000h]

CPA1_MEM_CONFIG_COMMIT is shown in Figure 3-275 and described in Table 3-273.

Return to the Summary Table.

Permament Lock for CPA1 Memory Configuration Register

Figure 3-275. CPA1_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CPA1_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-273. CPA1_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CPA1_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the CPA1_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset. 0 : CPA1_MEM_CONFIG is modifiable 1 : CPA1_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.4.7 CDA0_MEM_CONFIG Register (Offset = 20h) [Reset = 00000000h]

CDA0_MEM_CONFIG is shown in Figure 3-276 and described in Table 3-274.

Return to the Summary Table.

CDA0 Memory Configuration Register

Figure 3-276. CDA0_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-274. CDA0_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : CDA0 memory is not Intialized 1 : CDA0 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.4.8 CDA0_MEM_CONFIG_LOCK Register (Offset = 24h) [Reset = 00000000h]

CDA0_MEM_CONFIG_LOCK is shown in Figure 3-277 and described in Table 3-275.

Return to the Summary Table.

Temporary Lock for CDA0 Memory Configuration Register

Figure 3-277. CDA0_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA0_MEM_CONFIG
R-0h							R/W-0h

Table 3-275. CDA0_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA0_MEM_CONFIG	R/W	0h	0 : Write to CDA0_MEM_CONFIG is allowed. 1 : Write to CDA0_MEM_CONFIG is not allowed. Note : This bit can only be modified if CDA0_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.4.9 CDA0_MEM_CONFIG_COMMIT Register (Offset = 28h) [Reset = 00000000h]

CDA0_MEM_CONFIG_COMMIT is shown in Figure 3-278 and described in Table 3-276.

Return to the Summary Table.

Permament Lock for CDA0 Memory Configuration Register

Figure 3-278. CDA0_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA0_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-276. CDA0_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA0_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the CDA0_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset. 0 : CDA0_MEM_CONFIG is modifiable 1 : CDA0_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.4.10 CDA1_MEM_CONFIG Register (Offset = 30h) [Reset = 00000000h]

CDA1_MEM_CONFIG is shown in Figure 3-279 and described in Table 3-277.

Return to the Summary Table.

CDA1 Memory Configuration Register

Figure 3-279. CDA1_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-277. CDA1_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : CDA1 memory is not Intialized 1 : CDA1 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.4.11 CDA1_MEM_CONFIG_LOCK Register (Offset = 34h) [Reset = 00000000h]

CDA1_MEM_CONFIG_LOCK is shown in Figure 3-280 and described in Table 3-278.

Return to the Summary Table.

Temporary Lock for CDA1 Memory Configuration Register

Figure 3-280. CDA1_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA1_MEM_CONFIG
R-0h							R/W-0h

Table 3-278. CDA1_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA1_MEM_CONFIG	R/W	0h	0 : Write to CDA1_MEM_CONFIG is allowed. 1 : Write to CDA1_MEM_CONFIG is not allowed. Note : This bit can only be modified if CDA1_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.4.12 CDA1_MEM_CONFIG_COMMIT Register (Offset = 38h) [Reset = 00000000h]

CDA1_MEM_CONFIG_COMMIT is shown in Figure 3-281 and described in Table 3-279.

Return to the Summary Table.

Permament Lock for CDA1 Memory Configuration Register

Figure 3-281. CDA1_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA1_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-279. CDA1_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA1_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the CDA1_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset. 0 : CDA1_MEM_CONFIG is modifiable 1 : CDA1_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.4.13 CDA2_MEM_CONFIG Register (Offset = 40h) [Reset = 00000000h]

CDA2_MEM_CONFIG is shown in Figure 3-282 and described in Table 3-280.

Return to the Summary Table.

CDA2 Memory Configuration Register

Figure 3-282. CDA2_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-280. CDA2_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : CDA2 memory is not Intialized 1 : CDA2 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.4.14 CDA2_MEM_CONFIG_LOCK Register (Offset = 44h) [Reset = 00000000h]

CDA2_MEM_CONFIG_LOCK is shown in Figure 3-283 and described in Table 3-281.

Return to the Summary Table.

Temporary Lock for CDA2 Memory Configuration Register

Figure 3-283. CDA2_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA2_MEM_CONFIG
R-0h							R/W-0h

Table 3-281. CDA2_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA2_MEM_CONFIG	R/W	0h	0 : Write to CDA2_MEM_CONFIG is allowed. 1 : Write to CDA2_MEM_CONFIG is not allowed. Note : This bit can only be modified if CDA2_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.4.15 CDA2_MEM_CONFIG_COMMIT Register (Offset = 48h) [Reset = 00000000h]

CDA2_MEM_CONFIG_COMMIT is shown in Figure 3-284 and described in Table 3-282.

Return to the Summary Table.

Permament Lock for CDA2 Memory Configuration Register

Figure 3-284. CDA2_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA2_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-282. CDA2_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA2_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the CDA2_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset. 0 : CDA2_MEM_CONFIG is modifiable 1 : CDA2_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.4.16 CDA3_MEM_CONFIG Register (Offset = 50h) [Reset = 00000000h]

CDA3_MEM_CONFIG is shown in Figure 3-285 and described in Table 3-283.

Return to the Summary Table.

CDA3 Memory Configuration Register

Figure 3-285. CDA3_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						TESTMODE	
R-0h						R/W-0h	

Table 3-283. CDA3_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : CDA3 memory is not Intialized 1 : CDA3 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.4.17 CDA3_MEM_CONFIG_LOCK Register (Offset = 54h) [Reset = 00000000h]

CDA3_MEM_CONFIG_LOCK is shown in Figure 3-286 and described in Table 3-284.

Return to the Summary Table.

Temporary Lock for CDA3 Memory Configuration Register

Figure 3-286. CDA3_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA3_MEM_CONFIG
R-0h							R/W-0h

Table 3-284. CDA3_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA3_MEM_CONFIG	R/W	0h	0 : Write to CDA3_MEM_CONFIG is allowed. 1 : Write to CDA3_MEM_CONFIG is not allowed. Note : This bit can only be modified if CDA3_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.4.18 CDA3_MEM_CONFIG_COMMIT Register (Offset = 58h) [Reset = 00000000h]

CDA3_MEM_CONFIG_COMMIT is shown in Figure 3-287 and described in Table 3-285.

Return to the Summary Table.

Permament Lock for CDA3 Memory Configuration Register

Figure 3-287. CDA3_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA3_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-285. CDA3_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA3_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the CDA3_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset. 0 : CDA3_MEM_CONFIG is modifiable 1 : CDA3_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.4.19 CDA4_MEM_CONFIG Register (Offset = 60h) [Reset = 00000000h]

CDA4_MEM_CONFIG is shown in Figure 3-288 and described in Table 3-286.

Return to the Summary Table.

CDA4 Memory Configuration Register

Figure 3-288. CDA4_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-286. CDA4_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : CDA4 memory is not Intialized 1 : CDA4 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.4.20 CDA4_MEM_CONFIG_LOCK Register (Offset = 64h) [Reset = 00000000h]

CDA4_MEM_CONFIG_LOCK is shown in Figure 3-289 and described in Table 3-287.

Return to the Summary Table.

Temporary Lock for CDA4 Memory Configuration Register

Figure 3-289. CDA4_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA4_MEM_CONFIG
R-0h							R/W-0h

Table 3-287. CDA4_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA4_MEM_CONFIG	R/W	0h	0 : Write to CDA4_MEM_CONFIG is allowed. 1 : Write to CDA4_MEM_CONFIG is not allowed. Note : This bit can only be modified if CDA4_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.4.21 CDA4_MEM_CONFIG_COMMIT Register (Offset = 68h) [Reset = 00000000h]

CDA4_MEM_CONFIG_COMMIT is shown in Figure 3-290 and described in Table 3-288.

Return to the Summary Table.

Permament Lock for CDA4 Memory Configuration Register

Figure 3-290. CDA4_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA4_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-288. CDA4_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA4_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the CDA4_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset. 0 : CDA4_MEM_CONFIG is modifiable 1 : CDA4_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.4.22 CDA5_MEM_CONFIG Register (Offset = 70h) [Reset = 00000000h]

CDA5_MEM_CONFIG is shown in Figure 3-291 and described in Table 3-289.

Return to the Summary Table.

CDA5 Memory Configuration Register

Figure 3-291. CDA5_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						TESTMODE	
R-0h						R/W-0h	

Table 3-289. CDA5_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : CDA5 memory is not Intialized 1 : CDA5 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.4.23 CDA5_MEM_CONFIG_LOCK Register (Offset = 74h) [Reset = 00000000h]

CDA5_MEM_CONFIG_LOCK is shown in Figure 3-292 and described in Table 3-290.

Return to the Summary Table.

Temporary Lock for CDA5 Memory Configuration Register

Figure 3-292. CDA5_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA5_MEM_CONFIG
R-0h							R/W-0h

Table 3-290. CDA5_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA5_MEM_CONFIG	R/W	0h	0 : Write to CDA5_MEM_CONFIG is allowed. 1 : Write to CDA5_MEM_CONFIG is not allowed. Note : This bit can only be modified if CDA5_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.4.24 CDA5_MEM_CONFIG_COMMIT Register (Offset = 78h) [Reset = 00000000h]

CDA5_MEM_CONFIG_COMMIT is shown in Figure 3-293 and described in Table 3-291.

Return to the Summary Table.

Permament Lock for CDA5 Memory Configuration Register

Figure 3-293. CDA5_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA5_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-291. CDA5_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA5_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the CDA5_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset. 0 : CDA5_MEM_CONFIG is modifiable 1 : CDA5_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.4.25 CDA6_MEM_CONFIG Register (Offset = 80h) [Reset = 00000000h]

CDA6_MEM_CONFIG is shown in Figure 3-294 and described in Table 3-292.

Return to the Summary Table.

CDA6 Memory Configuration Register

Figure 3-294. CDA6_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-292. CDA6_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : CDA6 memory is not Intialized 1 : CDA6 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.4.26 CDA6_MEM_CONFIG_LOCK Register (Offset = 84h) [Reset = 00000000h]

CDA6_MEM_CONFIG_LOCK is shown in Figure 3-295 and described in Table 3-293.

Return to the Summary Table.

Temporary Lock for CDA6 Memory Configuration Register

Figure 3-295. CDA6_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA6_MEM_CONFIG
R-0h							R/W-0h

Table 3-293. CDA6_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA6_MEM_CONFIG	R/W	0h	0 : Write to CDA6_MEM_CONFIG is allowed. 1 : Write to CDA6_MEM_CONFIG is not allowed. Note : This bit can only be modified if CDA6_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.4.27 CDA6_MEM_CONFIG_COMMIT Register (Offset = 88h) [Reset = 00000000h]

CDA6_MEM_CONFIG_COMMIT is shown in Figure 3-296 and described in Table 3-294.

Return to the Summary Table.

Permament Lock for CDA6 Memory Configuration Register

Figure 3-296. CDA6_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA6_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-294. CDA6_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA6_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the CDA6_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset. 0 : CDA6_MEM_CONFIG is modifiable 1 : CDA6_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.4.28 CDA7_MEM_CONFIG Register (Offset = 90h) [Reset = 00000000h]

CDA7_MEM_CONFIG is shown in Figure 3-297 and described in Table 3-295.

Return to the Summary Table.

CDA7 Memory Configuration Register

Figure 3-297. CDA7_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-295. CDA7_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : CDA7 memory is not Intialized 1 : CDA7 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.4.29 CDA7_MEM_CONFIG_LOCK Register (Offset = 94h) [Reset = 00000000h]

CDA7_MEM_CONFIG_LOCK is shown in Figure 3-298 and described in Table 3-296.

Return to the Summary Table.

Temporary Lock for CDA7 Memory Configuration Register

Figure 3-298. CDA7_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA7_MEM_CONFIG
R-0h							R/W-0h

Table 3-296. CDA7_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA7_MEM_CONFIG	R/W	0h	0 : Write to CDA7_MEM_CONFIG is allowed. 1 : Write to CDA7_MEM_CONFIG is not allowed. Note : This bit can only be modified if CDA7_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.4.30 CDA7_MEM_CONFIG_COMMIT Register (Offset = 98h) [Reset = 00000000h]

CDA7_MEM_CONFIG_COMMIT is shown in Figure 3-299 and described in Table 3-297.

Return to the Summary Table.

Permament Lock for CDA7 Memory Configuration Register

Figure 3-299. CDA7_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA7_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-297. CDA7_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA7_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the CDA7_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset. 0 : CDA7_MEM_CONFIG is modifiable 1 : CDA7_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.4.31 CDA8_MEM_CONFIG Register (Offset = A0h) [Reset = 00000000h]

CDA8_MEM_CONFIG is shown in Figure 3-300 and described in Table 3-298.

Return to the Summary Table.

CDA8 Memory Configuration Register

Figure 3-300. CDA8_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-298. CDA8_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : CDA8 memory is not Intialized 1 : CDA8 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.4.32 CDA8_MEM_CONFIG_LOCK Register (Offset = A4h) [Reset = 00000000h]

CDA8_MEM_CONFIG_LOCK is shown in Figure 3-301 and described in Table 3-299.

Return to the Summary Table.

Temporary Lock for CDA8 Memory Configuration Register

Figure 3-301. CDA8_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA8_MEM_CONFIG
R-0h							R/W-0h

Table 3-299. CDA8_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA8_MEM_CONFIG	R/W	0h	0 : Write to CDA8_MEM_CONFIG is allowed. 1 : Write to CDA8_MEM_CONFIG is not allowed. Note : This bit can only be modified if CDA8_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.4.33 CDA8_MEM_CONFIG_COMMIT Register (Offset = A8h) [Reset = 00000000h]

CDA8_MEM_CONFIG_COMMIT is shown in Figure 3-302 and described in Table 3-300.

Return to the Summary Table.

Permament Lock for CDA8 Memory Configuration Register

Figure 3-302. CDA8_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA8_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-300. CDA8_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA8_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the CDA8_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset. 0 : CDA8_MEM_CONFIG is modifiable 1 : CDA8_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.4.34 CDA9_MEM_CONFIG Register (Offset = B0h) [Reset = 00000000h]

CDA9_MEM_CONFIG is shown in Figure 3-303 and described in Table 3-301.

Return to the Summary Table.

CDA9 Memory Configuration Register

Figure 3-303. CDA9_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						TESTMODE	
R-0h						R/W-0h	

Table 3-301. CDA9_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : CDA9 memory is not Intialized 1 : CDA9 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.4.35 CDA9_MEM_CONFIG_LOCK Register (Offset = B4h) [Reset = 00000000h]

CDA9_MEM_CONFIG_LOCK is shown in Figure 3-304 and described in Table 3-302.

Return to the Summary Table.

Temporary Lock for CDA9 Memory Configuration Register

Figure 3-304. CDA9_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA9_MEM_CONFIG
R-0h							R/W-0h

Table 3-302. CDA9_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA9_MEM_CONFIG	R/W	0h	0 : Write to CDA9_MEM_CONFIG is allowed. 1 : Write to CDA9_MEM_CONFIG is not allowed. Note : This bit can only be modified if CDA9_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.4.36 CDA9_MEM_CONFIG_COMMIT Register (Offset = B8h) [Reset = 00000000h]

CDA9_MEM_CONFIG_COMMIT is shown in Figure 3-305 and described in Table 3-303.

Return to the Summary Table.

Permament Lock for CDA9 Memory Configuration Register

Figure 3-305. CDA9_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA9_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-303. CDA9_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA9_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the CDA9_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset. 0 : CDA9_MEM_CONFIG is modifiable 1 : CDA9_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.4.37 CDA10_MEM_CONFIG Register (Offset = C0h) [Reset = 00000000h]

CDA10_MEM_CONFIG is shown in Figure 3-306 and described in Table 3-304.

Return to the Summary Table.

CDA10 Memory Configuration Register

Figure 3-306. CDA10_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						TESTMODE	
R-0h						R/W-0h	

Table 3-304. CDA10_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : CDA10 memory is not Intialized 1 : CDA10 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.4.38 CDA10_MEM_CONFIG_LOCK Register (Offset = C4h) [Reset = 00000000h]

CDA10_MEM_CONFIG_LOCK is shown in Figure 3-307 and described in Table 3-305.

Return to the Summary Table.

Temporary Lock for CDA10 Memory Configuration Register

Figure 3-307. CDA10_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA10_MEM_CONFIG
R-0h							R/W-0h

Table 3-305. CDA10_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA10_MEM_CONFIG	R/W	0h	0 : Write to CDA10_MEM_CONFIG is allowed. 1 : Write to CDA10_MEM_CONFIG is not allowed. Note : This bit can only be modified if CDA10_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.4.39 CDA10_MEM_CONFIG_COMMIT Register (Offset = C8h) [Reset = 00000000h]

CDA10_MEM_CONFIG_COMMIT is shown in Figure 3-308 and described in Table 3-306.

Return to the Summary Table.

Permament Lock for CDA10 Memory Configuration Register

Figure 3-308. CDA10_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA10_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-306. CDA10_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA10_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the CDA10_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset. 0 : CDA10_MEM_CONFIG is modifiable 1 : CDA10_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.4.40 CDA11_MEM_CONFIG Register (Offset = D0h) [Reset = 00000000h]

CDA11_MEM_CONFIG is shown in Figure 3-309 and described in Table 3-307.

Return to the Summary Table.

CDA11 Memory Configuration Register

Figure 3-309. CDA11_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TESTMODE
R-0h							R/W-0h

Table 3-307. CDA11_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : CDA11 memory is not Intialized 1 : CDA11 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.4.41 CDA11_MEM_CONFIG_LOCK Register (Offset = D4h) [Reset = 00000000h]

CDA11_MEM_CONFIG_LOCK is shown in Figure 3-310 and described in Table 3-308.

Return to the Summary Table.

Temporary Lock for CDA11 Memory Configuration Register

Figure 3-310. CDA11_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA11_MEM_CONFIG
R-0h							R/W-0h

Table 3-308. CDA11_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA11_MEM_CONFIG	R/W	0h	0 : Write to CDA11_MEM_CONFIG is allowed. 1 : Write to CDA11_MEM_CONFIG is not allowed. Note : This bit can only be modified if CDA11_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.4.42 CDA11_MEM_CONFIG_COMMIT Register (Offset = D8h) [Reset = 00000000h]

CDA11_MEM_CONFIG_COMMIT is shown in Figure 3-311 and described in Table 3-309.

Return to the Summary Table.

Permament Lock for CDA11 Memory Configuration Register

Figure 3-311. CDA11_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CDA11_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-309. CDA11_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CDA11_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the CDA11_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset. 0 : CDA11_MEM_CONFIG is modifiable 1 : CDA11_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.5 MEMSS_M_CONFIG_REGS Registers

Table 3-310 lists the memory-mapped registers for the MEMSS_M_CONFIG_REGS registers. All register offset addresses not listed in Table 3-310 should be considered as reserved locations and the register contents should not be modified.

Table 3-310. MEMSS_M_CONFIG_REGS Registers

Offset	Acronym	Register Name	Protection
0h	M0_MEM_CONFIG	M0 Memory Configuration Register	PARITY
4h	M0_MEM_CONFIG_LOCK	Temporary Lock for M0 Memory Configuration Register	PARITY
8h	M0_MEM_CONFIG_COMMIT	Permament Lock for M0 Memory Configuration Register	PARITY

Complex bit access types are encoded to fit into small table cells. Table 3-311 shows the codes that are used for access types in this section.

Table 3-311. MEMSS_M_CONFIG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value

3.13.5.1 M0_MEM_CONFIG Register (Offset = 0h) [Reset = 00000000h]

M0_MEM_CONFIG is shown in Figure 3-312 and described in Table 3-312.

Return to the Summary Table.

M0 Memory Configuration Register

Figure 3-312. M0_MEM_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							INIT_STS
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							INIT
R-0h							R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						TESTMODE	
R-0h						R/W-0h	

Table 3-312. M0_MEM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	INIT_STS	R	0h	0 : M0 memory is not Intialized 1 : M0 memory Initialization Done Reset type: CPU1.SYSRSn
23-17	RESERVED	R	0h	Reserved
16	INIT	R-0/W1S	0h	Writing '1' will trigger memory initialization. Reset type: CPU1.SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TESTMODE	R/W	0h	0 : Normal mode of operation 1 : Write to ECC bits is disabled. ECC check on Read Data is disabled. 2 : Write to Data bits is disabled. ECC check on Read Data is disabled.Read Returns ECC bits. 3 : Normal mode of operation Reset type: CPU1.SYSRSn

3.13.5.2 M0_MEM_CONFIG_LOCK Register (Offset = 4h) [Reset = 00000000h]

M0_MEM_CONFIG_LOCK is shown in Figure 3-313 and described in Table 3-313.

Return to the Summary Table.

Temporary Lock for M0 Memory Configuration Register

Figure 3-313. M0_MEM_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							M0_MEM_CONFIG
R-0h							R/W-0h

Table 3-313. M0_MEM_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	M0_MEM_CONFIG	R/W	0h	0 : Write to M0_MEM_CONFIG is allowed. 1 : Write to M0_MEM_CONFIG is not allowed. Note : This bit can only be modified if M0_MEM_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.5.3 M0_MEM_CONFIG_COMMIT Register (Offset = 8h) [Reset = 00000000h]

M0_MEM_CONFIG_COMMIT is shown in Figure 3-314 and described in Table 3-314.

Return to the Summary Table.

Permament Lock for M0 Memory Configuration Register

Figure 3-314. M0_MEM_CONFIG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							M0_MEM_CONFIG_LOCK
R-0h							WSonce-0h

Table 3-314. M0_MEM_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	M0_MEM_CONFIG_LOCK	WSonce	0h	When set, locks the M0_MEM_CONFIG_LOCK register. This bit cannot be cleared, except by reset. 0 : M0_MEM_CONFIG is modifiable 1 : M0_MEM_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.6 MEMSS_MISCI_REGS Registers

Table 3-315 lists the memory-mapped registers for the MEMSS_MISCI_REGS registers. All register offset addresses not listed in Table 3-315 should be considered as reserved locations and the register contents should not be modified.

Table 3-315. MEMSS_MISCI_REGS Registers

Offset	Acronym	Register Name	Protection
0h	MEM_DLB_CONFIG	Dataline buffer enable	PARITY
4h	MEM_DLB_CONFIG_LOCK	Temporary Lock for DLB Configuration Register	PARITY
8h	MEM_DLB_CONFIG_COMMIT	Permament Lock for DLB Configuration Register	PARITY
10h	PERI_MEM_TEST_LOCK	Peripheral Memory Test Lock Register	PARITY
14h	PERI_MEM_TEST_CONTROL	Peripheral Memory Test control Register	PARITY
1FCh	PARITY_TEST	Enabling the parity test feature	

Complex bit access types are encoded to fit into small table cells. Table 3-316 shows the codes that are used for access types in this section.

Table 3-316. MEMSS_MISCI_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.13.6.1 MEM_DLB_CONFIG Register (Offset = 0h) [Reset = 0000007Fh]

MEM_DLB_CONFIG is shown in Figure 3-315 and described in Table 3-317.

Return to the Summary Table.

Dataline buffer enable

Figure 3-315. MEM_DLB_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	SYNCBRIDGE_DLB_EN	RESERVED	RESERVED	RESERVED	CPU3_DLB_EN	CPU2_DLB_EN	CPU1_DLB_EN
R-0h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 3-317. MEM_DLB_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6	SYNCBRIDGE_DLB_EN	R/W	1h	0 : Disabled CPU6 dataline buffer 1 : Enable CPU6 dataline buffer Reset type: CPU1.SYSRSn
5	RESERVED	R/W	1h	Reserved
4	RESERVED	R/W	1h	Reserved
3	RESERVED	R/W	1h	Reserved
2	CPU3_DLB_EN	R/W	1h	0 : Disabled CPU3 dataline buffer 1 : Enable CPU3 dataline buffer Reset type: CPU1.SYSRSn
1	CPU2_DLB_EN	R/W	1h	0 : Disabled CPU2 dataline buffer 1 : Enable CPU2 dataline buffer Reset type: CPU1.SYSRSn
0	CPU1_DLB_EN	R/W	1h	0 : Disabled CPU1 dataline buffer 1 : Enable CPU1 dataline buffer Reset type: CPU1.SYSRSn

3.13.6.2 MEM_DLB_CONFIG_LOCK Register (Offset = 4h) [Reset = 00000000h]

MEM_DLB_CONFIG_LOCK is shown in Figure 3-316 and described in Table 3-318.

Return to the Summary Table.

Temporary Lock for DLB Configuration Register

Figure 3-316. MEM_DLB_CONFIG_LOCK Register

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

Table 3-318. MEM_DLB_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	MEM_DLB_CONFIG	R/W	0h	0 : Write to MEM_DLB_CONFIG is allowed. 1 : Write to MEM_DLB_CONFIG is not allowed. Note : This bit can only be modified if MEM_DLB_CONFIG_COMMIT.COMMIT is cleared. Reset type: CPU1.SYSRSn

3.13.6.3 MEM_DLB_CONFIG_COMMIT Register (Offset = 8h) [Reset = 00000000h]

MEM_DLB_CONFIG_COMMIT is shown in Figure 3-317 and described in Table 3-319.

Return to the Summary Table.

Permament Lock for DLB Configuration Register

Figure 3-317. MEM_DLB_CONFIG_COMMIT Register

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

Table 3-319. MEM_DLB_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	MEM_DLB_CONFIG_LOCK	WSonce	0h	When set, locks the MEM_DLB_CONFIG_LOCK register. This bit cannot be cleared, except by reset. 0 : MEM_DLB_CONFIG is modifiable 1 : MEM_DLB_CONFIG is committed permanently Reset type: CPU1.SYSRSn

3.13.6.4 PERI_MEM_TEST_LOCK Register (Offset = 10h) [Reset = 00000000h]

PERI_MEM_TEST_LOCK is shown in Figure 3-318 and described in Table 3-320.

Return to the Summary Table.

Peripheral Memory Test Lock Register

Figure 3-318. PERI_MEM_TEST_LOCK Register

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

Table 3-320. PERI_MEM_TEST_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PERI_MEM_TEST_CONTROL	R/W	0h	Locks write access to register PERI_MEM_TEST_CONTROL 0: Write access allowed 1: Write access blocked Reset type: CPU1.SYSRSn

3.13.6.5 PERI_MEM_TEST_CONTROL Register (Offset = 14h) [Reset = 00000000h]

PERI_MEM_TEST_CONTROL is shown in Figure 3-319 and described in Table 3-321.

Return to the Summary Table.

Peripheral Memory Test control Register

Figure 3-319. PERI_MEM_TEST_CONTROL Register

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

Table 3-321. PERI_MEM_TEST_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5	EtherCAT_MEM_FORCE_ERROR	R/W	0h	Force error bit 0 : No effect 1 : Parity bit going to Parity checker module of EtherCAT is inverted to introduce parity Error Reset type: CPU1.SYSRSn
4	RESERVED	R	0h	Reserved
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	RESERVED	R	0h	Reserved
0	RESERVED	R	0h	Reserved

3.13.6.6 PARITY_TEST Register (Offset = 1FCh) [Reset = 00000000h]

PARITY_TEST is shown in Figure 3-320 and described in Table 3-322.

Return to the Summary Table.

Enabling the parity test feature

Figure 3-320. PARITY_TEST Register

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

Table 3-322. PARITY_TEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	TESTEN	R/W	0h	1010: Parity test feature is enabled All other values: Parity test feature is disabled Note: (1) When the parity test feature is enabled, acutal registers are not accessible in the memory map. Instead, the parity values (final XOR output indicating the parity error) are accessible. Parity is computed for every byte and the correponding parity error value is available at the bit-0 of every byte. Value of '1' written to the parity bit after enabling the parity test feature can be used to inject the error by inverting the stored parity value. (2) It is recommended to leave the field as 0101 or 0000 after completing the parity test. Reset type: CPU1.SYSRSn

3.13.7 CPU_SYS_REGS Registers

Table 3-323 lists the memory-mapped registers for the CPU_SYS_REGS registers. All register offset addresses not listed in Table 3-323 should be considered as reserved locations and the register contents should not be modified.

Table 3-323. CPU_SYS_REGS Registers

Offset	Acronym	Register Name	Protection
0h	CPUSYSLOCK1	Lock bit for CPUSYS registers	PARITY
Ch	CPUID	Indicates CPU1, CPU2.. CPU6	PARITY
28h	LPMCR	LPM Control Register	PARITY
2Ch	CMPSSLPMSEL	CMPSS LPM Wakeup select registers	PARITY
30h	GPIOLPMSEL0	GPIO LPM Wakeup select registers	PARITY
34h	GPIOLPMSEL1	GPIO LPM Wakeup select registers	PARITY
38h	TMR2CLKCTL	Timer2 Clock Measurement functionality control register	PARITY
3Ch	RESCCLR	Reset Cause Clear Register	
40h	RESC	Reset Cause register	PARITY
70h	MCANWAKESTATUS	MCAN Wake Status Register	
74h	MCANWAKESTATUSCLR	MCAN Wake Status Clear Register	
78h	CLKSTOPREQ	Peripheral Clock Stop Request Register	PARITY
7Ch	CLKSTOPACK	Peripheral Clock Stop Ackonwledge Register	
80h	USER_REG1_SYSRSn	Software Configurable registers reset by SYSRSn	PARITY
84h	USER_REG2_SYSRSn	Software Configurable registers reset by SYSRSn	PARITY
88h	USER_REG1_XRSn	Software Configurable registers reset by XRSn	PARITY
8Ch	USER_REG2_XRSn	Software Configurable registers reset by XRSn	PARITY
90h	USER_REG1_PORESETn	Software Configurable registers reset by PORESETn	PARITY
94h	USER_REG2_PORESETn	Software Configurable registers reset by PORESETn	PARITY
98h	USER_REG3_PORESETn	Software Configurable registers reset by PORESETn	PARITY
9Ch	USER_REG4_PORESETn	Software Configurable registers reset by PORESETn	PARITY
A0h	JTAG_MMR_REG	Readback of JTAG registers for test purpose	
A4h	SIMRESET	Simulated Reset Register	PARITY
A8h	PARITY_TEST_ALT2	Enables parity test	

Complex bit access types are encoded to fit into small table cells. Table 3-324 shows the codes that are used for access types in this section.

Table 3-324. CPU_SYS_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
WSonce	W Sonce	Write Set once
Reset or Default Value		

Table 3-324. CPU_SYS_REGS Access Type Codes (continued)

Access Type	Code	Description
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.13.7.1 CPUSYSLOCK1 Register (Offset = 0h) [Reset = 00000000h]

CPUSYSLOCK1 is shown in Figure 3-321 and described in Table 3-325.

Return to the Summary Table.

Lock bit for CPUSYS registers
Notes:
[1] Any bit in this register, once set can only be cleared through a CPU1.SYSRSn. Write of 0 to any bit of this register has no effect
[2] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed

Figure 3-321. CPUSYSLOCK1 Register

31	30	29	28	27	26	25	24
USER_REG4_PORESETn	USER_REG3_PORESETn	USER_REG2_PORESETn	USER_REG1_PORESETn	USER_REG2_XRSn	USER_REG1_XRSn	USER_REG2_SYSRSn	USER_REG1_SYSRSn
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		CMPSSLPMSEL	RESERVED	GPIOLPMSEL1	GPIOLPMSEL0	LPMCR	RESERVED
R-0-0h		R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 3-325. CPUSYSLOCK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	USER_REG4_PORESETn	R/WSonce	0h	Lock bit for USER_REG4_PORESETn Register 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
30	USER_REG3_PORESETn	R/WSonce	0h	Lock bit for USER_REG3_PORESETn Register 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
29	USER_REG2_PORESETn	R/WSonce	0h	Lock bit for USER_REG2_PORESETn Register 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
28	USER_REG1_PORESETn	R/WSonce	0h	Lock bit for USER_REG1_PORESETn Register 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
27	USER_REG2_XRSn	R/WSonce	0h	Lock bit for USER_REG2_XRSn Register 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
26	USER_REG1_XRSn	R/WSonce	0h	Lock bit for USER_REG1_XRSn Register 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn

Table 3-325. CPUSYSLOCK1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	USER_REG2_SYSRSn	R/WSonce	0h	Lock bit for USER_REG2_SYSRSn Register 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
24	USER_REG1_SYSRSn	R/WSonce	0h	Lock bit for USER_REG1_SYSRSn Register 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
23-6	RESERVED	R-0	0h	Reserved
5	CMPSSLPMSEL	R/WSonce	0h	Lock bit for CMPSSLPMSEL Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
4	RESERVED	R/WSonce	0h	Reserved
3	GPIOLPMSEL1	R/WSonce	0h	Lock bit for GPIOLPMSEL1 Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
2	GPIOLPMSEL0	R/WSonce	0h	Lock bit for GPIOLPMSEL0 Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
1	LPMCR	R/WSonce	0h	Lock bit for LPMCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
0	RESERVED	R/WSonce	0h	Reserved

3.13.7.2 CPUID Register (Offset = Ch) [Reset = 00000001h]

CPUID is shown in Figure 3-322 and described in Table 3-326.

Return to the Summary Table.

Indicates CPU1, CPU2.. CPU6

Figure 3-322. CPUID Register

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

Table 3-326. CPUID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1-0	CPUID	R	1h	CPUID = 1 for CPU1, 2 for CPU2, 3 for CPU3, 4 for CPU4, 5 for CPU5, 6 for CPU6 Reset type: SYSRSn

3.13.7.3 LPMCR Register (Offset = 28h) [Reset = 000000FCh]

LPMCR is shown in Figure 3-323 and described in Table 3-327.

Return to the Summary Table.

LPM Control Register

Figure 3-323. LPMCR Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED						
R/W1S-0h	R-0-0h						
23	22	21	20	19	18	17	16
RESERVED						RESERVED	
R-0-0h						R/W-0h	
15	14	13	12	11	10	9	8
WDINTE	RESERVED						
R/W-0h	R-0-0h						
7	6	5	4	3	2	1	0
QUALSTDBY						LPM	
R/W-3Fh						R/W-0h	

Table 3-327. LPMCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W1S	0h	Reserved
30-18	RESERVED	R-0	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15	WDINTE	R/W	0h	When this bit is set to 1, it enables the watchdog interrupt signal to wake the device from STANDBY mode. Note: [1] To use this signal, the user must also enable the WDINTn signal using the WDENINT bit in the SCSR register. This signal will not wake the device from HALT mode because the clock to watchdog module is turned off Reset type: SYSRSn
14-8	RESERVED	R-0	0h	Reserved
7-2	QUALSTDBY	R/W	3Fh	Select number of OSCCLK clock cycles to qualify the selected inputs when waking the from STANDBY mode: 000000 = 2 OSCCLKs 000001 = 3 OSCCLKs ..... 111111 = 65 OSCCLKs Note: The LPMCR.QUALSTDBY register should be set to a value greater than the ratio of INTOSC1/PLLSYSCLK to ensure proper wake up. Reset type: SYSRSn
1-0	LPM	R/W	0h	These bits set the low power mode for the device. Takes effect when CPU executes the IDLE instruction (when IDLE instruction is out of EXE Phase of the Pipeline) 00: IDLE Mode 01: STANDBY Mode 1x: STANDBY Mode Reset type: SYSRSn

3.13.7.4 CMPSSLPMSEL Register (Offset = 2Ch) [Reset = 00000000h]

CMPSSLPMSEL is shown in Figure 3-324 and described in Table 3-328.

Return to the Summary Table.

CMPSS LPM Wakeup select registers

Figure 3-324. CMPSSLPMSEL Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
CMPSS12L	CMPSS12H	CMPSS11L	CMPSS11H	CMPSS10L	CMPSS10H	CMPSS9L	CMPSS9H
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
CMPSS8L	CMPSS8H	CMPSS7L	CMPSS7H	CMPSS6L	CMPSS6H	CMPSS5L	CMPSS5H
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CMPSS4L	CMPSS4H	CMPSS3L	CMPSS3H	CMPSS2L	CMPSS2H	CMPSS1L	CMPSS1H
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-328. CMPSSLPMSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	CMPSS12L	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
22	CMPSS12H	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
21	CMPSS11L	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
20	CMPSS11H	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
19	CMPSS10L	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
18	CMPSS10H	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn

Table 3-328. CMPSSLPMSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	CMPSS9L	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
16	CMPSS9H	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
15	CMPSS8L	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
14	CMPSS8H	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
13	CMPSS7L	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
12	CMPSS7H	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
11	CMPSS6L	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
10	CMPSS6H	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
9	CMPSS5L	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
8	CMPSS5H	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
7	CMPSS4L	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
6	CMPSS4H	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
5	CMPSS3L	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
4	CMPSS3H	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
3	CMPSS2L	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
2	CMPSS2H	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
1	CMPSS1L	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn

Table 3-328. CMPSSLPMSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	CMPSS1H	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn

3.13.7.5 GPIOLPMSEL0 Register (Offset = 30h) [Reset = 00000000h]

GPIOLPMSEL0 is shown in Figure 3-325 and described in Table 3-329.

Return to the Summary Table.

GPIO LPM Wakeup select registers

Figure 3-325. GPIOLPMSEL0 Register

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-329. GPIOLPMSEL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO31	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
30	GPIO30	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
29	GPIO29	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
28	GPIO28	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
27	GPIO27	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
26	GPIO26	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
25	GPIO25	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
24	GPIO24	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
23	GPIO23	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
22	GPIO22	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn

Table 3-329. GPIOLPMSEL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	GPIO21	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
20	GPIO20	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
19	GPIO19	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
18	GPIO18	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
17	GPIO17	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
16	GPIO16	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
15	GPIO15	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
14	GPIO14	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
13	GPIO13	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
12	GPIO12	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
11	GPIO11	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
10	GPIO10	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
9	GPIO9	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
8	GPIO8	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
7	GPIO7	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
6	GPIO6	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
5	GPIO5	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn

Table 3-329. GPIOLPMSEL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	GPIO4	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
3	GPIO3	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
2	GPIO2	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
1	GPIO1	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
0	GPIO0	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn

3.13.7.6 GPIOLPMSEL1 Register (Offset = 34h) [Reset = 00000000h]

GPIOLPMSEL1 is shown in Figure 3-326 and described in Table 3-330.

Return to the Summary Table.

GPIO LPM Wakeup select registers

Figure 3-326. GPIOLPMSEL1 Register

31	30	29	28	27	26	25	24
GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-330. GPIOLPMSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO63	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
30	GPIO62	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
29	GPIO61	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
28	GPIO60	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
27	GPIO59	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
26	GPIO58	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
25	GPIO57	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
24	GPIO56	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
23	GPIO55	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
22	GPIO54	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn

Table 3-330. GPIOLPMSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	GPIO53	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
20	GPIO52	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
19	GPIO51	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
18	GPIO50	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
17	GPIO49	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
16	GPIO48	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
15	GPIO47	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
14	GPIO46	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
13	GPIO45	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
12	GPIO44	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
11	GPIO43	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
10	GPIO42	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
9	GPIO41	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
8	GPIO40	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
7	GPIO39	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
6	GPIO38	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
5	GPIO37	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn

Table 3-330. GPIOLPMSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	GPIO36	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
3	GPIO35	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
2	GPIO34	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
1	GPIO33	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
0	GPIO32	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn

3.13.7.7 TMR2CLKCTL Register (Offset = 38h) [Reset = 00000000h]

TMR2CLKCTL is shown in Figure 3-327 and described in Table 3-331.

Return to the Summary Table.

Timer2 Clock Measurement functionality control register

Figure 3-327. TMR2CLKCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED			TMR2CLKPRESCALE			TMR2CLKSRCSEL	
R-0-0h			R/W-0h			R/W-0h	

Table 3-331. TMR2CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5-3	TMR2CLKPRESCALE	R/W	0h	CPU Timer 2 Clock Pre-Scale Value: These bits select the pre-scale value for the selected clock source for CPU Timer 2: 0,0,0,/1 (default on reset) 0,0,1,/2, 0,1,0,/4 0,1,1,/8 1,0,0,/16 1,0,1,spare (defaults to /16) 1,1,0,spare (defaults to /16) 1,1,1,spare (defaults to /16) Note: [1] The CPU Timer2s Clock sync logic detects an input clock edge when configured for any clock source other than SYSCLK and generates an appropriate clock pulse to the CPU timer2. If SYSCLK is approximately the same or less then the input clock source, then the user would need to configure the pre-scale value such that SYSCLK is at least twice as fast as the pre-scaled value. Reset type: SYSRSn
2-0	TMR2CLKSRCSEL	R/W	0h	CPU Timer 2 Clock Source Select Bit: This bit selects the source for CPU Timer 2: 000 = PLLSYSCLK Selected (default on reset, pre-scale is bypassed) 001 = INTOSC1 010 = INTOSC2 011 = XTAL 100 = FLC1_PUMPOSC 101 = FLC2_PUMPOSC 110 = AUXPLLCLK (Reserved) 111 = CRUDEOSC Reset type: SYSRSn

3.13.7.8 RESCCLR Register (Offset = 3Ch) [Reset = 00000000h]

RESCCLR is shown in Figure 3-328 and described in Table 3-332.

Return to the Summary Table.

Reset Cause Clear Register

Figure 3-328. RESCCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED			ESMXRSn	SIMRESET_XRSn	RESERVED	ECAT_RESET_OUT	RESERVED
R-0-0h			W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1S-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	NMIWDRSn	WDRSn	XRSn	POR
R-0-0h	W1S-0h	W1S-0h	R-0-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h

Table 3-332. RESCCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R-0	0h	Reserved
12	ESMXRSn	W1C	0h	Clear bit for corresponding status bit in RESC. Read of RESCCLR always gives 0. Writing a 1 to this bit clears the status bit in RESC to 0 Writing 0 has no effect. Reset type: SYSRSn
11	SIMRESET_XRSn	W1C	0h	Clear bit for corresponding status bit in RESC. Read of RESCCLR always gives 0. Writing a 1 to this bit clears the status bit in RESC to 0 Writing 0 has no effect. Reset type: SYSRSn
10	RESERVED	W1C	0h	Reserved
9	ECAT_RESET_OUT	W1C	0h	Clear bit for corresponding status bit in RESC. Read of RESCCLR always gives 0. Writing a 1 to this bit clears the status bit in RESC to 0 Writing 0 has no effect. Reset type: SYSRSn
8	RESERVED	W1S	0h	Reserved
7	RESERVED	R-0	0h	Reserved
6	RESERVED	W1S	0h	Reserved
5	RESERVED	W1S	0h	Reserved
4	RESERVED	R-0	0h	Reserved
3	NMIWDRSn	W1S	0h	Clear bit for corresponding status bit in RESC. Read of RESCCLR always gives 0. Writing a 1 to this bit clears the status bit in RESC to 0 Writing 0 has no effect. Reset type: SYSRSn

Table 3-332. RESCCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	WDRSn	W1S	0h	Clear bit for corresponding status bit in RESC. Read of RESCCLR always gives 0. Writing a 1 to this bit clears the status bit in RESC to 0 Writing 0 has no effect. Reset type: SYSRSn
1	XRSn	W1S	0h	Clear bit for corresponding status bit in RESC. Read of RESCCLR always gives 0. Writing a 1 to this bit clears the status bit in RESC to 0 Writing 0 has no effect. Reset type: SYSRSn
0	POR	W1S	0h	Clear bit for corresponding status bit in RESC. Read of RESCCLR always gives 0. Writing a 1 to this bit clears the status bit in RESC to 0 Writing 0 has no effect. Reset type: SYSRSn

3.13.7.9 RESC Register (Offset = 40h) [Reset = X0000003h]

RESC is shown in Figure 3-329 and described in Table 3-333.

Return to the Summary Table.

Reset Cause register

Figure 3-329. RESC Register

31	30	29	28	27	26	25	24
DCON	XRSn_pin_status	RESERVED					
R-0h	R-Xh	R-0-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED			ESMRESET	SIMRESET_XRSn	RESERVED	ECAT_RESET_OUT	RESERVED
R-0-0h			R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	NMIWDRSn	WDRSn	XRSn	POR
R-0-0h	R-0h	R-0h	R-0-0h	R-0h	R-0h	R-1h	R-1h

Table 3-333. RESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DCON	R	0h	Reading this bit provides the status of debugger connection to the CPU. 0 : Debugger is not connected to the CPU 1 : Debugger is connected to the CPU Notes: [1] This bit is connected to the DCON o/p signal of the CPU Reset type: N/A
30	XRSn_pin_status	R	Xh	Reading this bit provides the current status of the XRSn pin. Reset value is reflective of the pin status. Reset type: N/A
29-13	RESERVED	R-0	0h	Reserved
12	ESMRESET	R	0h	If this bit is set, indicates that the device was reset from ESM, SYS_ESM tile (Critical Priority Interrupt output of SYS_ESM) Note: To know the exact cause of NMI after the reset, software needs to read ESM registers Reset type: PORESETn
11	SIMRESET_XRSn	R	0h	If this bit is set, indicates that the device was reset by SIMRESET_XRSn Reset type: PORESETn
10	RESERVED	R	0h	Reserved
9	ECAT_RESET_OUT	R	0h	If this bit is set, indicates that the device was reset by ECAT_RESET_OUT Writing a 1 to this bit will force the bit to 0 Writing of 0 will have no effect. Reset type: PORESETn
8	RESERVED	R	0h	Reserved
7	RESERVED	R-0	0h	Reserved
6	RESERVED	R	0h	Reserved
5	RESERVED	R	0h	Reserved

Table 3-333. RESC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	RESERVED	R-0	0h	Reserved
3	NMIWDRSn	R	0h	If this bit is set, indicates that the device was reset by NMIWDRSn (Issued from ESM, CPU specific tile High Priority Interrupt output of ESM) Note: To know the exact cause of NMI after the reset, software needs to read ESM registers Reset type: PORESETn
2	WDRSn	R	0h	If this bit is set, indicates that the device was reset by WDRSn. Note: [1] A bit inside WD module also provides the same information. This bit is present to keep things consistent. This register is a one-stop shop for the software to know the reset cause for the C29x core. Reset type: PORESETn
1	XRSn	R	1h	If this bit is set, indicates that the device was reset by XRSn. Reset type: PORESETn
0	POR	R	1h	If this bit is set, indicates that the device was reset by PORn. Reset type: PORESETn

3.13.7.10 MCANWAKESTATUS Register (Offset = 70h) [Reset = 00000000h]

MCANWAKESTATUS is shown in Figure 3-330 and described in Table 3-334.

Return to the Summary Table.

MCAN Wake Status Register

Figure 3-330. MCANWAKESTATUS Register

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

Table 3-334. MCANWAKESTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5	WAKE_MCANF	R	0h	0 : wakeup event has not occured. 1 : wakeup event has occured. Reset type: CPUx.SYSRSn
4	WAKE_MCANE	R	0h	0 : wakeup event has not occured. 1 : wakeup event has occured. Reset type: CPUx.SYSRSn
3	WAKE_MCAND	R	0h	0 : wakeup event has not occured. 1 : wakeup event has occured. Reset type: CPUx.SYSRSn
2	WAKE_MCANC	R	0h	0 : wakeup event has not occured. 1 : wakeup event has occured. Reset type: CPUx.SYSRSn
1	WAKE_MCANB	R	0h	0 : wakeup event has not occured. 1 : wakeup event has occured. Reset type: CPUx.SYSRSn
0	WAKE_MCANA	R	0h	0 : wakeup event has not occured. 1 : wakeup event has occured. Reset type: CPUx.SYSRSn

3.13.7.11 MCANWAKESTATUSCLR Register (Offset = 74h) [Reset = 00000000h]

MCANWAKESTATUSCLR is shown in Figure 3-331 and described in Table 3-335.

Return to the Summary Table.

MCAN Wake Status Clear Register

Figure 3-331. MCANWAKESTATUSCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		WAKE_MCANF	WAKE_MCANE	WAKE_MCAND	WAKE_MCANC	WAKE_MCANB	WAKE_MCANA
R-0h		R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 3-335. MCANWAKESTATUSCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5	WAKE_MCANF	R-0/W1S	0h	MCANF 0 : No effect. 1 : Clears WAKE_MCANF bit of MCANWAKESTATUS register Reset type: CPUx.SYSRSn
4	WAKE_MCANE	R-0/W1S	0h	MCANE 0 : No effect. 1 : Clears WAKE_MCANE bit of MCANWAKESTATUS register Reset type: CPUx.SYSRSn
3	WAKE_MCAND	R-0/W1S	0h	MCAND 0 : No effect. 1 : Clears WAKE_MCAND bit of MCANWAKESTATUS register Reset type: CPUx.SYSRSn
2	WAKE_MCANC	R-0/W1S	0h	MCANC 0 : No effect. 1 : Clears WAKE_MCANC bit of MCANWAKESTATUS register Reset type: CPUx.SYSRSn
1	WAKE_MCANB	R-0/W1S	0h	MCANB 0 : No effect. 1 : Clears WAKE_MCANB bit of MCANWAKESTATUS register Reset type: CPUx.SYSRSn
0	WAKE_MCANA	R-0/W1S	0h	MCANA 0 : No effect. 1 : Clears WAKE_MCANA bit of MCANWAKESTATUS register Reset type: CPUx.SYSRSn

3.13.7.12 CLKSTOPREQ Register (Offset = 78h) [Reset = 00000000h]

CLKSTOPREQ is shown in Figure 3-332 and described in Table 3-336.

Return to the Summary Table.

Peripheral Clock Stop Request Register

Figure 3-332. CLKSTOPREQ Register

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

Table 3-336. CLKSTOPREQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write to any of the bits in this register will succeed only if a value of 0x5634 is written to the KEY field. Reset type: CPUx.SYSRSn
15-14	RESERVED	R-0	0h	Reserved
13	MCAN_F	R/W	0h	MCAN_F Clock Stop Request Bit 0: If clock to MCAN_F is turned off, it will be turned on, else no effect. 1: Clock stop request toMCAN_F Note: Once set, this bit is cleared when clock to MCAN_F is turned on as a result of a wakeup event in hardware Reset type: CPUx.SYSRSn
12	MCAN_E	R/W	0h	MCAN_E Clock Stop Request Bit 0: If clock to MCAN_E is turned off, it will be turned on, else no effect. 1: Clock stop request toMCAN_E Note: Once set, this bit is cleared when clock to MCAN_E is turned on as a result of a wakeup event in hardware Reset type: CPUx.SYSRSn
11	MCAN_D	R/W	0h	MCAN_D Clock Stop Request Bit 0: If clock to MCAN_D is turned off, it will be turned on, else no effect. 1: Clock stop request toMCAN_D Note: Once set, this bit is cleared when clock to MCAN_D is turned on as a result of a wakeup event in hardware Reset type: CPUx.SYSRSn
10	MCAN_C	R/W	0h	MCAN_C Clock Stop Request Bit 0: If clock to MCAN_C is turned off, it will be turned on, else no effect. 1: Clock stop request toMCAN_C Note: Once set, this bit is cleared when clock to MCAN_C is turned on as a result of a wakeup event in hardware Reset type: CPUx.SYSRSn

Table 3-336. CLKSTOPREQ Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	MCAN_B	R/W	0h	MCAN_B Clock Stop Request Bit 0: If clock to MCAN_B is turned off, it will be turned on, else no effect. 1: Clock stop request toMCAN_B Note: Once set, this bit is cleared when clock to MCAN_B is turned on as a result of a wakeup event in hardware Reset type: CPUx.SYSRSn
8	MCAN_A	R/W	0h	MCAN_A Clock Stop Request Bit 0: If clock to MCAN_A is turned off, it will be turned on, else no effect. 1: Clock stop request toMCAN_A Note: Once set, this bit is cleared when clock to MCAN_A is turned on as a result of a wakeup event in hardware Reset type: CPUx.SYSRSn
7-6	RESERVED	R-0	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R-0	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R-0	0h	Reserved
0	RESERVED	R/W	0h	Reserved

3.13.7.13 CLKSTOPACK Register (Offset = 7Ch) [Reset = 00000000h]

CLKSTOPACK is shown in Figure 3-333 and described in Table 3-337.

Return to the Summary Table.

Peripheral Clock Stop Ackonwledge Register

Figure 3-333. CLKSTOPACK Register

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

Table 3-337. CLKSTOPACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	Reserved
13	MCAN_F	R	0h	MCAN_F Clock Stop Acknowledge Bit 0: Clock stop request not acknowledged 1: Clock stop acknowledged Reset type: CPUx.SYSRSn
12	MCAN_E	R	0h	MCAN_E Clock Stop Acknowledge Bit 0: Clock stop request not acknowledged 1: Clock stop acknowledged Reset type: CPUx.SYSRSn
11	MCAN_D	R	0h	MCAN_D Clock Stop Acknowledge Bit 0: Clock stop request not acknowledged 1: Clock stop acknowledged Reset type: CPUx.SYSRSn
10	MCAN_C	R	0h	MCAN_C Clock Stop Acknowledge Bit 0: Clock stop request not acknowledged 1: Clock stop acknowledged Reset type: CPUx.SYSRSn
9	MCAN_B	R	0h	MCAN_B Clock Stop Acknowledge Bit 0: Clock stop request not acknowledged 1: Clock stop acknowledged Reset type: CPUx.SYSRSn
8	MCAN_A	R	0h	MCAN_A Clock Stop Acknowledge Bit 0: Clock stop request not acknowledged 1: Clock stop acknowledged Reset type: CPUx.SYSRSn
7-6	RESERVED	R-0	0h	Reserved
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	RESERVED	R-0	0h	Reserved
2	RESERVED	R	0h	Reserved

Table 3-337. CLKSTOPACK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	RESERVED	R-0	0h	Reserved
0	RESERVED	R	0h	Reserved

3.13.7.14 USER_REG1_SYSRSn Register (Offset = 80h) [Reset = 00000000h]

USER_REG1_SYSRSn is shown in Figure 3-334 and described in Table 3-338.

Return to the Summary Table.

Software Configurable registers reset by SYSRSn

Figure 3-334. USER_REG1_SYSRSn Register

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

Table 3-338. USER_REG1_SYSRSn Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITS	R/W	0h	R/W bits reset by SYSRSn to be used by the application software Reset type: SYSRSn

3.13.7.15 USER_REG2_SYSRSn Register (Offset = 84h) [Reset = 00000000h]

USER_REG2_SYSRSn is shown in Figure 3-335 and described in Table 3-339.

Return to the Summary Table.

Software Configurable registers reset by SYSRSn

Figure 3-335. USER_REG2_SYSRSn Register

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

Table 3-339. USER_REG2_SYSRSn Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITS	R/W	0h	R/W bits reset by SYSRSn to be used by the application software Reset type: SYSRSn

3.13.7.16 USER_REG1_XRSn Register (Offset = 88h) [Reset = 00000000h]

USER_REG1_XRSn is shown in Figure 3-336 and described in Table 3-340.

Return to the Summary Table.

Software Configurable registers reset by XRSn

Figure 3-336. USER_REG1_XRSn Register

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

Table 3-340. USER_REG1_XRSn Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITS	R/W	0h	R/W bits reset by XRSn to be used by the application software Reset type: XRSn

3.13.7.17 USER_REG2_XRSn Register (Offset = 8Ch) [Reset = 00000000h]

USER_REG2_XRSn is shown in Figure 3-337 and described in Table 3-341.

Return to the Summary Table.

Software Configurable registers reset by XRSn

Figure 3-337. USER_REG2_XRSn Register

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

Table 3-341. USER_REG2_XRSn Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITS	R/W	0h	R/W bits reset by XRSn to be used by the application software Reset type: XRSn

3.13.7.18 USER_REG1_PORESETn Register (Offset = 90h) [Reset = 00000000h]

USER_REG1_PORESETn is shown in Figure 3-338 and described in Table 3-342.

Return to the Summary Table.

Software Configurable registers reset by PORESETn

Figure 3-338. USER_REG1_PORESETn Register

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

Table 3-342. USER_REG1_PORESETn Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITS	R/W	0h	R/W bits reset by PORESETn to be used by the application software Reset type: PORESETn

3.13.7.19 USER_REG2_PORESETn Register (Offset = 94h) [Reset = 00000000h]

USER_REG2_PORESETn is shown in Figure 3-339 and described in Table 3-343.

Return to the Summary Table.

Software Configurable registers reset by PORESETn

Figure 3-339. USER_REG2_PORESETn Register

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

Table 3-343. USER_REG2_PORESETn Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITS	R/W	0h	R/W bits reset by PORESETn to be used by the application software Reset type: PORESETn

3.13.7.20 USER_REG3_PORESETn Register (Offset = 98h) [Reset = 00000000h]

USER_REG3_PORESETn is shown in Figure 3-340 and described in Table 3-344.

Return to the Summary Table.

Software Configurable registers reset by PORESETn

Figure 3-340. USER_REG3_PORESETn Register

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

Table 3-344. USER_REG3_PORESETn Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITS	R/W	0h	R/W bits reset by PORESETn to be used by the application software Reset type: PORESETn

3.13.7.21 USER_REG4_PORESETn Register (Offset = 9Ch) [Reset = 00000000h]

USER_REG4_PORESETn is shown in Figure 3-341 and described in Table 3-345.

Return to the Summary Table.

Software Configurable registers reset by PORESETn

Figure 3-341. USER_REG4_PORESETn Register

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

Table 3-345. USER_REG4_PORESETn Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITS	R/W	0h	R/W bits reset by PORESETn to be used by the application software Reset type: PORESETn

3.13.7.22 JTAG_MMR_REG Register (Offset = A0h) [Reset = 00000000h]

JTAG_MMR_REG is shown in Figure 3-342 and described in Table 3-346.

Return to the Summary Table.

Readback of JTAG registers for test purpose

Figure 3-342. JTAG_MMR_REG Register

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

Table 3-346. JTAG_MMR_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

3.13.7.23 SIMRESET Register (Offset = A4h) [Reset = 00000000h]

SIMRESET is shown in Figure 3-343 and described in Table 3-347.

Return to the Summary Table.

Only for CPU1

Figure 3-343. SIMRESET Register

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
KEY							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						XRSn	RESERVED
R-0-0h						R-0/W1S-0h	R-0/W1S-0h

Table 3-347. SIMRESET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write to this register succeeds only if this field is written with a value of 0xa5a5 Note: [1] Due to this KEY, only 32-bit writes will succeed (provided the KEY matches). 16-bit writes to the upper or lower half of this register will be ignored Reset type: XRSn
15-2	RESERVED	R-0	0h	Reserved
1	XRSn	R-0/W1S	0h	Writing a 1 to this field generates a XRSn like reset. Writing a 0 has no effect. Note: Writing to this pin will pull the XRSn pin low for 512 INTOSC1 clock cycles. This bit is implemented only for SIMRESET copy of CPU1. For CPU2,3,4,5,6 this is a reserved bit. No effect on writes, and Read=0 Reset type: XRSn
0	RESERVED	R-0/W1S	0h	Reserved

3.13.7.24 PARITY_TEST_ALT2 Register (Offset = A8h) [Reset = 00000000h]

PARITY_TEST_ALT2 is shown in Figure 3-344 and described in Table 3-348.

Return to the Summary Table.

Enables parity test

Figure 3-344. PARITY_TEST_ALT2 Register

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

Table 3-348. PARITY_TEST_ALT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	TESTEN	R/W	0h	1010: Parity test feature is enabled All other values: Parity test feature is disabled Note: (1) When the parity test feature is enabled, actual registers are not accessible in the memory map. Instead, the parity values (final XOR output indicating the parity error) are accessible. Parity is computed for every byte and the corresponding parity error value is available at the bit-0 of every byte. Value of '1' written to the parity bit after enabling the parity test feature can be used to inject the error by inverting the stored parity value. (2) It is recommended to leave the field as 0101 or 0000 after completing the parity test. Reset type: SYSRSn

3.13.8 CPU_PER_CFG_REGS Registers

Table 3-349 lists the memory-mapped registers for the CPU_PER_CFG_REGS registers. All register offset addresses not listed in Table 3-349 should be considered as reserved locations and the register contents should not be modified.

Table 3-349. CPU_PER_CFG_REGS Registers

Offset	Acronym	Register Name	Protection
0h	CPUPERCFGLOCK1	Lock bit for CPUx_PER_CFG registers	PARITY
4h	CPUPERCFGLOCK2	Lock bit for CPUx_PER_CFG registers	PARITY
10h	PCLKCR0	Peripheral Clock Gating Registers	PARITY
14h	PCLKCR1	Peripheral Clock Gating Register - EMIF	PARITY
18h	PCLKCR2	Peripheral Clock Gating Register - EPWM	PARITY
1Ch	PCLKCR3	Peripheral Clock Gating Register - ECAP	PARITY
20h	PCLKCR4	Peripheral Clock Gating Register - EQEP	PARITY
28h	PCLKCR6	Peripheral Clock Gating Register - SDFM	PARITY
2Ch	PCLKCR7	Peripheral Clock Gating Register - SCI, UART	PARITY
30h	PCLKCR8	Peripheral Clock Gating Register - SPI	PARITY
34h	PCLKCR9	Peripheral Clock Gating Register - I2C	PARITY
38h	PCLKCR10	Peripheral Clock Gating Register - CAN	PARITY
44h	PCLKCR13	Peripheral Clock Gating Register - ADC	PARITY
48h	PCLKCR14	Peripheral Clock Gating Register - CMPSS	PARITY
50h	PCLKCR16	Peripheral Clock Gating Register Buf_DAC	PARITY
54h	PCLKCR17	Peripheral Clock Gating Register - CLB	PARITY
58h	PCLKCR18	Peripheral Clock Gating Register - FSI	PARITY
5Ch	PCLKCR19	Peripheral Clock Gating Register - LIN	PARITY
60h	PCLKCR20	Peripheral Clock Gating Register - PMBUS	PARITY
64h	PCLKCR21	Peripheral Clock Gating Register - DCC	PARITY
6Ch	PCLKCR23	Peripheral Clock Gating Register - EtherCAT	PARITY
74h	PCLKCR25	Peripheral Clock Gating Register - HRCAL0,1,2	PARITY
7Ch	PCLKCR27	Peripheral Clock Gating Register - EPG	PARITY
80h	PCLKCR28	Peripheral Clock Gating Register - ADCCHECKER	PARITY
88h	PCLKCR30	Peripheral Clock Gating Register - SENT	PARITY
90h	PCLKCR32	Peripheral Clock Gating Register - WADI	PARITY
110h	SOFTPRES0	Processing Block Software Reset register	PARITY
114h	SOFTPRES1	EMIF Software Reset register	PARITY
118h	SOFTPRES2	EPWM Software Reset register	PARITY
11Ch	SOFTPRES3	ECAP Software Reset register	PARITY
120h	SOFTPRES4	EQEP Software Reset register	PARITY
128h	SOFTPRES6	Sigma Delta Software Reset register	PARITY
12Ch	SOFTPRES7	SCI, UART Software Reset register	PARITY
130h	SOFTPRES8	SPI Software Reset register	PARITY
134h	SOFTPRES9	I2C Software Reset register	PARITY
138h	SOFTPRES10	CAN Software Reset register	PARITY
144h	SOFTPRES13	ADC Software Reset register	PARITY
148h	SOFTPRES14	CMPSS Software Reset register	PARITY
150h	SOFTPRES16	DAC Software Reset register	PARITY
154h	SOFTPRES17	CLB Software Reset register	PARITY

Table 3-349. CPU_PER_CFG_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
158h	SOFTPRES18	FSI Software Reset register	PARITY
15Ch	SOFTPRES19	LIN Software Reset register	PARITY
160h	SOFTPRES20	PMBUS Software Reset register	PARITY
164h	SOFTPRES21	DCC Software Reset register	PARITY
16Ch	SOFTPRES23	ETHERCAT Software Reset register	PARITY
174h	SOFTPRES25	HRCAL0,1,2 Software Reset register	PARITY
17Ch	SOFTPRES27	EPG Software Reset register	PARITY
180h	SOFTPRES28	ADCCHECKER Software Reset register	PARITY
188h	SOFTPRES30	SENT Software Reset register	PARITY
190h	SOFTPRES32	WADI Software Reset register	PARITY
1B8h	PARITY_TEST_ALT1	Enables parity test	

Complex bit access types are encoded to fit into small table cells. Table 3-350 shows the codes that are used for access types in this section.

Table 3-350. CPU_PER_CFG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.13.8.1 CPUPERCFGLOCK1 Register (Offset = 0h) [Reset = 00000000h]

CPUPERCFGLOCK1 is shown in Figure 3-345 and described in Table 3-351.

Return to the Summary Table.

Lock bit for CPUx_PER_CFG registers
Notes:
[1] Any bit in this register, once set can only be cleared through a CPUx.SYSRSn. Write of 0 to any bit of this register has no effect
[2] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed

Figure 3-345. CPUPERCFGLOCK1 Register

31	30	29	28	27	26	25	24
RESERVED	PCLKCR30	RESERVED	PCLKCR28	PCLKCR27	RESERVED	PCLKCR25	RESERVED
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
PCLKCR23	RESERVED	PCLKCR21	PCLKCR20	PCLKCR19	PCLKCR18	PCLKCR17	PCLKCR16
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
RESERVED	PCLKCR14	PCLKCR13	RESERVED	RESERVED	PCLKCR10	PCLKCR9	PCLKCR8
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
PCLKCR7	PCLKCR6	RESERVED	PCLKCR4	PCLKCR3	PCLKCR2	PCLKCR1	PCLKCR0
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 3-351. CPUPERCFGLOCK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/WSonce	0h	Reserved
30	PCLKCR30	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
29	RESERVED	R/WSonce	0h	Reserved
28	PCLKCR28	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
27	PCLKCR27	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
26	RESERVED	R/WSonce	0h	Reserved
25	PCLKCR25	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
24	RESERVED	R/WSonce	0h	Reserved
23	PCLKCR23	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
22	RESERVED	R/WSonce	0h	Reserved

Table 3-351. CPUPERCFGLOCK1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	PCLKCR21	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
20	PCLKCR20	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
19	PCLKCR19	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
18	PCLKCR18	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
17	PCLKCR17	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
16	PCLKCR16	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
15	RESERVED	R/WSonce	0h	Reserved
14	PCLKCR14	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
13	PCLKCR13	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
12	RESERVED	R/WSonce	0h	Reserved
11	RESERVED	R/WSonce	0h	Reserved
10	PCLKCR10	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
9	PCLKCR9	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
8	PCLKCR8	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
7	PCLKCR7	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
6	PCLKCR6	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn

Table 3-351. CPUPERCFGLOCK1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	RESERVED	R/WSonce	0h	Reserved
4	PCLKCR4	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
3	PCLKCR3	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
2	PCLKCR2	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
1	PCLKCR1	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn
0	PCLKCR0	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn

3.13.8.2 CPUPERCFGLOCK2 Register (Offset = 4h) [Reset = 00000000h]

CPUPERCFGLOCK2 is shown in Figure 3-346 and described in Table 3-352.

Return to the Summary Table.

Lock bit for CPUx_PER_CFG registers

Figure 3-346. CPUPERCFGLOCK2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							PCLKCR32
R-0-0h							R/WSonce-0h

Table 3-352. CPUPERCFGLOCK2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	PCLKCR32	R/WSonce	0h	Lock bit for PCLKCR Register: 0: Respective register is not locked 1: Respective register is locked. Reset type: CPUx.SYSRSn

3.13.8.3 PCLKCR0 Register (Offset = 10h) [Reset = 0300007Ch]

PCLKCR0 is shown in Figure 3-347 and described in Table 3-353.

Return to the Summary Table.

Peripheral Clock Gating Registers

Figure 3-347. PCLKCR0 Register

31	30	29	28	27	26	25	24
RESERVED						CPUx_DLT	CPUx_ERAD
R-0-0h						R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
RESERVED			GTBCLKSYNC	TBCLKSYNC	RESERVED		
R-0-0h			R/W-0h	R/W-0h	R-0-0h		
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	CPUTIMER2	CPUTIMER1	CPUTIMER0	RTDMA2	RTDMA1	RESERVED	RESERVED
R-0-0h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R-0-0h	R-0-0h

Table 3-353. PCLKCR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	Reserved
25	CPUx_DLT	R/W	1h	DLT Clock Enable Bit: When set, this enables the clock to the DLT module respective to the CPUx 1: DLT clock is enabled 0: DLT clock is disabled Reset type: SYSRSn
24	CPUx_ERAD	R/W	1h	ERAD Clock Enable Bit: When set, this enables the clock to the ERAD module respective to the CPUx 1: ERAD clock is enabled 0: ERAD clock is disabled Reset type: SYSRSn
23-21	RESERVED	R-0	0h	Reserved
20	GTBCLKSYNC	R/W	0h	EPWM Time Base Clock Global sync: When set by CPU1, PWM time bases of all modules start counting. The effect of this bit is seen on all the EPWM modules irrespective of their partitioning based on CPUSEL Notes: 1. This bit on the CPUx.PCLKCR0 9x=2,3 etc.] register has no effect. 2. Writing '1' to this bit overrides the effect of write '1' to the TBCLKSYNC bit at the same time Reset type: SYSRSn
19	TBCLKSYNC	R/W	0h	EPWM Time Base Clock sync: When set PWM time bases of all the PWM modules belonging to the same CPU-Subsystem (as partitioned using their CPUSEL bits) start counting Reset type: SYSRSn
18-7	RESERVED	R-0	0h	Reserved
6	CPUTIMER2	R/W	1h	Module Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

Table 3-353. PCLKCR0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	CPUTIMER1	R/W	1h	Module Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
4	CPUTIMER0	R/W	1h	Module Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
3	RTDMA2	R/W	1h	Module Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
2	RTDMA1	R/W	1h	Module Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
1	RESERVED	R-0	0h	Reserved
0	RESERVED	R-0	0h	Reserved

3.13.8.4 PCLKCR1 Register (Offset = 14h) [Reset = 00000000h]

PCLKCR1 is shown in Figure 3-348 and described in Table 3-354.

Return to the Summary Table.

Peripheral Clock Gating Register - EMIF

Figure 3-348. PCLKCR1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	EMIF1
R-0-0h						R/W-0h	R/W-0h

Table 3-354. PCLKCR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	EMIF1	R/W	0h	EMIF1 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Notes: [1] These bits are not used (R/W) in CPU2/CPU3.PCLKCR1 register. EMIF1 clock enabled are controlled only from CPU1.PCLKCR1 register. Reset type: SYSRSn

3.13.8.5 PCLKCR2 Register (Offset = 18h) [Reset = 00000000h]

PCLKCR2 is shown in Figure 3-349 and described in Table 3-355.

Return to the Summary Table.

Peripheral Clock Gating Register - EPWM

Figure 3-349. PCLKCR2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						EPWM18	EPWM17
R-0-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
EPWM16	EPWM15	EPWM14	EPWM13	EPWM12	EPWM11	EPWM10	EPWM9
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
EPWM8	EPWM7	EPWM6	EPWM5	EPWM4	EPWM3	EPWM2	EPWM1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-355. PCLKCR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	Reserved
17	EPWM18	R/W	0h	EPWM18 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
16	EPWM17	R/W	0h	EPWM17 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
15	EPWM16	R/W	0h	EPWM16 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
14	EPWM15	R/W	0h	EPWM15 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
13	EPWM14	R/W	0h	EPWM14 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
12	EPWM13	R/W	0h	EPWM13 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
11	EPWM12	R/W	0h	EPWM12 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

Table 3-355. PCLKCR2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	EPWM11	R/W	0h	EPWM11 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
9	EPWM10	R/W	0h	EPWM10 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
8	EPWM9	R/W	0h	EPWM9 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
7	EPWM8	R/W	0h	EPWM8 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
6	EPWM7	R/W	0h	EPWM7 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
5	EPWM6	R/W	0h	EPWM6 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
4	EPWM5	R/W	0h	EPWM5 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
3	EPWM4	R/W	0h	EPWM4 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
2	EPWM3	R/W	0h	EPWM3 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
1	EPWM2	R/W	0h	EPWM2 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
0	EPWM1	R/W	0h	EPWM1 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.6 PCLKCR3 Register (Offset = 1Ch) [Reset = 00000000h]

PCLKCR3 is shown in Figure 3-350 and described in Table 3-356.

Return to the Summary Table.

Peripheral Clock Gating Register - ECAP

Figure 3-350. PCLKCR3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	ECAP6	ECAP5	ECAP4	ECAP3	ECAP2	ECAP1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-356. PCLKCR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	ECAP6	R/W	0h	ECAP6 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
4	ECAP5	R/W	0h	ECAP5 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
3	ECAP4	R/W	0h	ECAP4 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
2	ECAP3	R/W	0h	ECAP3 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
1	ECAP2	R/W	0h	ECAP2 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
0	ECAP1	R/W	0h	ECAP1 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.7 PCLKCR4 Register (Offset = 20h) [Reset = 00000000h]

PCLKCR4 is shown in Figure 3-351 and described in Table 3-357.

Return to the Summary Table.

Peripheral Clock Gating Register - EQEP

Figure 3-351. PCLKCR4 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		EQEP6	EQEP5	EQEP4	EQEP3	EQEP2	EQEP1
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-357. PCLKCR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	EQEP6	R/W	0h	EQEP4 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
4	EQEP5	R/W	0h	EQEP3 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
3	EQEP4	R/W	0h	EQEP4 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
2	EQEP3	R/W	0h	EQEP3 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
1	EQEP2	R/W	0h	EQEP2 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
0	EQEP1	R/W	0h	EQEP1 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.8 PCLKCR6 Register (Offset = 28h) [Reset = 00000000h]

PCLKCR6 is shown in Figure 3-352 and described in Table 3-358.

Return to the Summary Table.

Peripheral Clock Gating Register - SDFM

Figure 3-352. PCLKCR6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED	RESERVED	RESERVED	RESERVED	SD4	SD3	SD2	SD1
R-0-0h								R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-358. PCLKCR6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	SD4	R/W	0h	SD4 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
2	SD3	R/W	0h	SD3 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
1	SD2	R/W	0h	SD2 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
0	SD1	R/W	0h	SD1 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.9 PCLKCR7 Register (Offset = 2Ch) [Reset = 00000000h]

PCLKCR7 is shown in Figure 3-353 and described in Table 3-359.

Return to the Summary Table.

Peripheral Clock Gating Register - SCI, UART

Figure 3-353. PCLKCR7 Register

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

Table 3-359. PCLKCR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R-0	0h	Reserved
21	UART_F	R/W	0h	Module Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
20	UART_E	R/W	0h	Module Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
19	UART_D	R/W	0h	Module Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
18	UART_C	R/W	0h	Module Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
17	UART_B	R/W	0h	Module Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
16	UART_A	R/W	0h	Module Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
15-4	RESERVED	R-0	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved

Table 3-359. PCLKCR7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	RESERVED	R/W	0h	Reserved

3.13.8.10 PCLKCR8 Register (Offset = 30h) [Reset = 00000000h]

PCLKCR8 is shown in Figure 3-354 and described in Table 3-360.

Return to the Summary Table.

Peripheral Clock Gating Register - SPI

Figure 3-354. PCLKCR8 Register

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

Table 3-360. PCLKCR8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15-5	RESERVED	R-0	0h	Reserved
4	SPI_E	R/W	0h	SPI_D Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
3	SPI_D	R/W	0h	SPI_D Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
2	SPI_C	R/W	0h	SPI_C Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
1	SPI_B	R/W	0h	SPI_B Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
0	SPI_A	R/W	0h	SPI_A Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.11 PCLKCR9 Register (Offset = 34h) [Reset = 00000000h]

PCLKCR9 is shown in Figure 3-355 and described in Table 3-361.

Return to the Summary Table.

Peripheral Clock Gating Register - I2C

Figure 3-355. PCLKCR9 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						I2C_B	I2C_A
R-0-0h						R/W-0h	R/W-0h

Table 3-361. PCLKCR9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	I2C_B	R/W	0h	I2C_B Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
0	I2C_A	R/W	0h	I2C_A Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.12 PCLKCR10 Register (Offset = 38h) [Reset = 00000000h]

PCLKCR10 is shown in Figure 3-356 and described in Table 3-362.

Return to the Summary Table.

Peripheral Clock Gating Register - CAN

Figure 3-356. PCLKCR10 Register

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

Table 3-362. PCLKCR10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R-0	0h	Reserved
9	MCAN_F	R/W	0h	MCAN_D Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
8	MCAN_E	R/W	0h	MCAN_C Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
7	MCAN_D	R/W	0h	MCAN_D Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
6	MCAN_C	R/W	0h	MCAN_C Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
5	MCAN_B	R/W	0h	MCAN_B Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
4	MCAN_A	R/W	0h	MCAN_A Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

3.13.8.13 PCLKCR13 Register (Offset = 44h) [Reset = 00000000h]

PCLKCR13 is shown in Figure 3-357 and described in Table 3-363.

Return to the Summary Table.

Peripheral Clock Gating Register - ADC

Figure 3-357. PCLKCR13 Register

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

Table 3-363. PCLKCR13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R-0	0h	Reserved
4	ADC_E	R/W	0h	ADC_D Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
3	ADC_D	R/W	0h	ADC_D Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
2	ADC_C	R/W	0h	ADC_C Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
1	ADC_B	R/W	0h	ADC_B Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
0	ADC_A	R/W	0h	ADC_A Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.14 PCLKCR14 Register (Offset = 48h) [Reset = 00000000h]

PCLKCR14 is shown in Figure 3-358 and described in Table 3-364.

Return to the Summary Table.

Peripheral Clock Gating Register - CMPSS

Figure 3-358. PCLKCR14 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				CMPSS12	CMPSS11	CMPSS10	CMPSS9
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CMPSS8	CMPSS7	CMPSS6	CMPSS5	CMPSS4	CMPSS3	CMPSS2	CMPSS1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-364. PCLKCR14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11	CMPSS12	R/W	0h	CMPSS12 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
10	CMPSS11	R/W	0h	CMPSS11 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
9	CMPSS10	R/W	0h	CMPSS10 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
8	CMPSS9	R/W	0h	CMPSS9 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
7	CMPSS8	R/W	0h	CMPSS8 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
6	CMPSS7	R/W	0h	CMPSS7 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
5	CMPSS6	R/W	0h	CMPSS6 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

Table 3-364. PCLKCR14 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CMPSS5	R/W	0h	CMPSS5 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
3	CMPSS4	R/W	0h	CMPSS4 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
2	CMPSS3	R/W	0h	CMPSS3 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
1	CMPSS2	R/W	0h	CMPSS2 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
0	CMPSS1	R/W	0h	CMPSS1 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.15 PCLKCR16 Register (Offset = 50h) [Reset = 00000000h]

PCLKCR16 is shown in Figure 3-359 and described in Table 3-365.

Return to the Summary Table.

Peripheral Clock Gating Register Buf_DAC

Figure 3-359. PCLKCR16 Register

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

Table 3-365. PCLKCR16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	DAC_B	R/W	0h	Buffered_DAC_B Clock Enable Bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
16	DAC_A	R/W	0h	Buffered_DAC_A Clock Enable Bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
15-4	RESERVED	R-0	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

3.13.8.16 PCLKCR17 Register (Offset = 54h) [Reset = 00000000h]

PCLKCR17 is shown in Figure 3-360 and described in Table 3-366.

Return to the Summary Table.

Peripheral Clock Gating Register - CLB

Figure 3-360. PCLKCR17 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		CLB6	CLB5	CLB4	CLB3	CLB2	CLB1
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-366. PCLKCR17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	CLB6	R/W	0h	CLB4 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
4	CLB5	R/W	0h	CLB3 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
3	CLB4	R/W	0h	CLB4 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
2	CLB3	R/W	0h	CLB3 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
1	CLB2	R/W	0h	CLB2 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
0	CLB1	R/W	0h	CLB1 Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.17 PCLKCR18 Register (Offset = 58h) [Reset = 00000000h]

PCLKCR18 is shown in Figure 3-361 and described in Table 3-367.

Return to the Summary Table.

Peripheral Clock Gating Register - FSI

Figure 3-361. PCLKCR18 Register

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

Table 3-367. PCLKCR18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	Reserved
19	FSIRX_D	R/W	0h	FSIRX_C Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
18	FSIRX_C	R/W	0h	FSITX_C Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
17	FSIRX_B	R/W	0h	FSIRX_B Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
16	FSIRX_A	R/W	0h	FSITX_B Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
15-4	RESERVED	R-0	0h	Reserved
3	FSITX_D	R/W	0h	FSIRX_A Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
2	FSITX_C	R/W	0h	FSITX_A Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
1	FSITX_B	R/W	0h	FSIRX_A Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

Table 3-367. PCLKCR18 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	FSITX_A	R/W	0h	FSITX_A Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.18 PCLKCR19 Register (Offset = 5Ch) [Reset = 00000000h]

PCLKCR19 is shown in Figure 3-362 and described in Table 3-368.

Return to the Summary Table.

Peripheral Clock Gating Register - LIN

Figure 3-362. PCLKCR19 Register

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

Table 3-368. PCLKCR19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	LIN_B	R/W	0h	LIN_B Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
0	LIN_A	R/W	0h	LIN_A Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.19 PCLKCR20 Register (Offset = 60h) [Reset = 00000000h]

PCLKCR20 is shown in Figure 3-363 and described in Table 3-369.

Return to the Summary Table.

Peripheral Clock Gating Register - PMBUS

Figure 3-363. PCLKCR20 Register

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

Table 3-369. PCLKCR20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	PMBUS_A	R/W	0h	PMBUS_A Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.20 PCLKCR21 Register (Offset = 64h) [Reset = 00000000h]

PCLKCR21 is shown in Figure 3-364 and described in Table 3-370.

Return to the Summary Table.

Peripheral Clock Gating Register - DCC

Figure 3-364. PCLKCR21 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					DCC3	DCC2	DCC1
R-0-0h					R/W-0h	R/W-0h	R/W-0h

Table 3-370. PCLKCR21 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	DCC3	R/W	0h	DCC Clock Enable Bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
1	DCC2	R/W	0h	DCC Clock Enable Bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
0	DCC1	R/W	0h	DCC Clock Enable Bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.21 PCLKCR23 Register (Offset = 6Ch) [Reset = 00000000h]

PCLKCR23 is shown in Figure 3-365 and described in Table 3-371.

Return to the Summary Table.

Peripheral Clock Gating Register - EtherCAT

Figure 3-365. PCLKCR23 Register

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

Table 3-371. PCLKCR23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	ETHERCAT	R/W	0h	ETHERCAT Clock Enable Bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.22 PCLKCR25 Register (Offset = 74h) [Reset = 00000000h]

PCLKCR25 is shown in Figure 3-366 and described in Table 3-372.

Return to the Summary Table.

To be implemented only for CPU1

Figure 3-366. PCLKCR25 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					HRCAL2	HRCAL1	HRCAL0
R-0-0h					R/W-0h	R/W-0h	R/W-0h

Table 3-372. PCLKCR25 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	HRCAL2	R/W	0h	HRCA2 Clock Enable Bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
1	HRCAL1	R/W	0h	HRCAL1 Clock Enable Bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
0	HRCAL0	R/W	0h	HRCAL0 Clock Enable Bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.23 PCLKCR27 Register (Offset = 7Ch) [Reset = 00000000h]

PCLKCR27 is shown in Figure 3-367 and described in Table 3-373.

Return to the Summary Table.

Peripheral Clock Gating Register - EPG

Figure 3-367. PCLKCR27 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							EPG1
R-0-0h							R/W-0h

Table 3-373. PCLKCR27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	EPG1	R/W	0h	EPG1 Clock Enable Bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.24 PCLKCR28 Register (Offset = 80h) [Reset = 00000000h]

PCLKCR28 is shown in Figure 3-368 and described in Table 3-374.

Return to the Summary Table.

Peripheral Clock Gating Register - ADCCHECKER

Figure 3-368. PCLKCR28 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED		RESERVED	RESERVED	RESERVED	ADCSEAGGRCPU3	ADCSEAGGRCPU2	ADCSEAGGRCPU1
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED						ADCCHECKER10	ADCCHECKER9
R-0-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
ADCCHECKER8	ADCCHECKER7	ADCCHECKER6	ADCCHECKER5	ADCCHECKER4	ADCCHECKER3	ADCCHECKER2	ADCCHECKER1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-374. PCLKCR28 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R-0	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	ADCSEAGGRCPU3	R/W	0h	Clock Enable bit fro ADC Safety Checker Error Aggegator module for CPU1: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
17	ADCSEAGGRCPU2	R/W	0h	Clock Enable bit fro ADC Safety Checker Error Aggegator module for CPU2: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
16	ADCSEAGGRCPU1	R/W	0h	Clock Enable bit fro ADC Safety Checker Error Aggegator module for CPU1: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
15-10	RESERVED	R-0	0h	Reserved
9	ADCCHECKER10	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
8	ADCCHECKER9	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

Table 3-374. PCLKCR28 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	ADCCHECKER8	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
6	ADCCHECKER7	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
5	ADCCHECKER6	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
4	ADCCHECKER5	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
3	ADCCHECKER4	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
2	ADCCHECKER3	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
1	ADCCHECKER2	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
0	ADCCHECKER1	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.25 PCLKCR30 Register (Offset = 88h) [Reset = 00000000h]

PCLKCR30 is shown in Figure 3-369 and described in Table 3-375.

Return to the Summary Table.

Peripheral Clock Gating Register - SENT

Figure 3-369. PCLKCR30 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		SENT6	SENT5	SENT4	SENT3	SENT2	SENT1
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-375. PCLKCR30 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	SENT6	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
4	SENT5	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
3	SENT4	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
2	SENT3	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
1	SENT2	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
0	SENT1	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.26 PCLKCR32 Register (Offset = 90h) [Reset = 00000000h]

PCLKCR32 is shown in Figure 3-370 and described in Table 3-376.

Return to the Summary Table.

Peripheral Clock Gating Register - WADI

Figure 3-370. PCLKCR32 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						WADI2	WADI1
R-0-0h						R/W-0h	R/W-0h

Table 3-376. PCLKCR32 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	WADI2	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn
0	WADI1	R/W	0h	Clock Enable bit: 0: Module clock is gated-off 1: Module clock is turned-on Reset type: SYSRSn

3.13.8.27 SOFTPRES0 Register (Offset = 110h) [Reset = 00000000h]

SOFTPRES0 is shown in Figure 3-371 and described in Table 3-377.

Return to the Summary Table.

Processing Block Software Reset register
When bits in this register are set, the respective module is in reset. All design data is lost and the module registers are returned to their reset states. Bits must be manually cleared after being set.

Figure 3-371. SOFTPRES0 Register

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

Table 3-377. SOFTPRES0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	Reserved
25	CPUx_DLT	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
24	CPUx_ERAD	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
23-7	RESERVED	R-0	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	CPUTIMER	R/W	0h	1: Modules is under reset 0: Modules reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
3-0	RESERVED	R-0	0h	Reserved

3.13.8.28 SOFTPRES1 Register (Offset = 114h) [Reset = 00000000h]

SOFTPRES1 is shown in Figure 3-372 and described in Table 3-378.

Return to the Summary Table.

Peripheral Software Reset register
When bits in this register are set, the respective peripheral is in reset. All data is lost and the peripheral registers are returned to their reset states. Bits must be manually cleared after being set.

Figure 3-372. SOFTPRES1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	EMIF1
R-0-0h						R/W-0h	R/W-0h

Table 3-378. SOFTPRES1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	EMIF1	R/W	0h	When this bit is set, only the control logic of the respective EMIF1 is reset. It does not reset the internal registers except the Total Access register and the Total Activate register. Refer to EMIF spec for more details on the EMIF SOFTRESET feature. This bit must be manually cleared after being set. 1: EMIF1 is under SOFTRESET 0: Module reset is determined by the device Reset Network Notes: [1] These bits are not used (R/W) in CPU2/CPU3.SOFTPRES1 register. EMIF1 clock enabled are controlled only from CPU1.SOFTPRES1 register. Reset type: CPUx.SYSRSn

3.13.8.29 SOFTPRES2 Register (Offset = 118h) [Reset = 00000000h]

SOFTPRES2 is shown in Figure 3-373 and described in Table 3-379.

Return to the Summary Table.

EPWM Software Reset register

Figure 3-373. SOFTPRES2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						EPWM18	EPWM17
R-0-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
EPWM16	EPWM15	EPWM14	EPWM13	EPWM12	EPWM11	EPWM10	EPWM9
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
EPWM8	EPWM7	EPWM6	EPWM5	EPWM4	EPWM3	EPWM2	EPWM1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-379. SOFTPRES2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	Reserved
17	EPWM18	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
16	EPWM17	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
15	EPWM16	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
14	EPWM15	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
13	EPWM14	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
12	EPWM13	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
11	EPWM12	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
10	EPWM11	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
9	EPWM10	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

Table 3-379. SOFTPRES2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	EPWM9	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
7	EPWM8	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
6	EPWM7	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
5	EPWM6	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
4	EPWM5	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
3	EPWM4	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
2	EPWM3	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
1	EPWM2	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
0	EPWM1	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.30 SOFTPRES3 Register (Offset = 11Ch) [Reset = 00000000h]

SOFTPRES3 is shown in Figure 3-374 and described in Table 3-380.

Return to the Summary Table.

ECAP Software Reset register

Figure 3-374. SOFTPRES3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	ECAP6	ECAP5	ECAP4	ECAP3	ECAP2	ECAP1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-380. SOFTPRES3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	ECAP6	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
4	ECAP5	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
3	ECAP4	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
2	ECAP3	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
1	ECAP2	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
0	ECAP1	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.31 SOFTPRES4 Register (Offset = 120h) [Reset = 00000000h]

SOFTPRES4 is shown in Figure 3-375 and described in Table 3-381.

Return to the Summary Table.

EQEP Software Reset register

Figure 3-375. SOFTPRES4 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		EQEP6	EQEP5	EQEP4	EQEP3	EQEP2	EQEP1
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-381. SOFTPRES4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	EQEP6	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
4	EQEP5	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
3	EQEP4	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
2	EQEP3	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
1	EQEP2	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
0	EQEP1	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.32 SOFTPRES6 Register (Offset = 128h) [Reset = 00000000h]

SOFTPRES6 is shown in Figure 3-376 and described in Table 3-382.

Return to the Summary Table.

Sigma Delta Software Reset register

Figure 3-376. SOFTPRES6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED	RESERVED	RESERVED	RESERVED	SD4	SD3	SD2	SD1
R-0-0h								R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-382. SOFTPRES6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	SD4	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
2	SD3	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
1	SD2	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
0	SD1	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.33 SOFTPRES7 Register (Offset = 12Ch) [Reset = 00000000h]

SOFTPRES7 is shown in Figure 3-377 and described in Table 3-383.

Return to the Summary Table.

SCI, UART Software Reset register

Figure 3-377. SOFTPRES7 Register

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

Table 3-383. SOFTPRES7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R-0	0h	Reserved
21	UART_F	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
20	UART_E	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
19	UART_D	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
18	UART_C	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
17	UART_B	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
16	UART_A	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
15-4	RESERVED	R-0	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

3.13.8.34 SOFTPRES8 Register (Offset = 130h) [Reset = 00000000h]

SOFTPRES8 is shown in Figure 3-378 and described in Table 3-384.

Return to the Summary Table.

SPI Software Reset register

Figure 3-378. SOFTPRES8 Register

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

Table 3-384. SOFTPRES8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15-5	RESERVED	R-0	0h	Reserved
4	SPI_E	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
3	SPI_D	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
2	SPI_C	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
1	SPI_B	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
0	SPI_A	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.35 SOFTPRES9 Register (Offset = 134h) [Reset = 00000000h]

SOFTPRES9 is shown in Figure 3-379 and described in Table 3-385.

Return to the Summary Table.

I2C Software Reset register

Figure 3-379. SOFTPRES9 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						I2C_B	I2C_A
R-0-0h						R/W-0h	R/W-0h

Table 3-385. SOFTPRES9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	I2C_B	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
0	I2C_A	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.36 SOFTPRES10 Register (Offset = 138h) [Reset = 00000XX0h]

SOFTPRES10 is shown in Figure 3-380 and described in Table 3-386.

Return to the Summary Table.

CAN Software Reset register

Figure 3-380. SOFTPRES10 Register

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

Table 3-386. SOFTPRES10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R-0	0h	Reserved
9	MCAN_F	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
8	MCAN_E	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
7	MCAN_D	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
6	MCAN_C	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
5	MCAN_B	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
4	MCAN_A	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

3.13.8.37 SOFTPRES13 Register (Offset = 144h) [Reset = 00000000h]

SOFTPRES13 is shown in Figure 3-381 and described in Table 3-387.

Return to the Summary Table.

ADC Software Reset register

Figure 3-381. SOFTPRES13 Register

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

Table 3-387. SOFTPRES13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	ADC_E	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
3	ADC_D	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
2	ADC_C	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
1	ADC_B	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
0	ADC_A	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.38 SOFTPRES14 Register (Offset = 148h) [Reset = 00000000h]

SOFTPRES14 is shown in Figure 3-382 and described in Table 3-388.

Return to the Summary Table.

CMPSS Software Reset register

Figure 3-382. SOFTPRES14 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				CMPSS12	CMPSS11	CMPSS10	CMPSS9
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CMPSS8	CMPSS7	CMPSS6	CMPSS5	CMPSS4	CMPSS3	CMPSS2	CMPSS1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-388. SOFTPRES14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11	CMPSS12	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
10	CMPSS11	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
9	CMPSS10	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
8	CMPSS9	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
7	CMPSS8	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
6	CMPSS7	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
5	CMPSS6	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
4	CMPSS5	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
3	CMPSS4	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

Table 3-388. SOFTPRES14 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	CMPSS3	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
1	CMPSS2	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
0	CMPSS1	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.39 SOFTPRES16 Register (Offset = 150h) [Reset = 00000000h]

SOFTPRES16 is shown in Figure 3-383 and described in Table 3-389.

Return to the Summary Table.

DAC Software Reset register

Figure 3-383. SOFTPRES16 Register

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

Table 3-389. SOFTPRES16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	DAC_B	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
16	DAC_A	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
15-4	RESERVED	R-0	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

3.13.8.40 SOFTPRES17 Register (Offset = 154h) [Reset = 0000000Xh]

SOFTPRES17 is shown in Figure 3-384 and described in Table 3-390.

Return to the Summary Table.

CLB Software Reset register

Figure 3-384. SOFTPRES17 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		CLB6	CLB5	CLB4	CLB3	CLB2	CLB1
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-Xh	R-Xh

Table 3-390. SOFTPRES17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5	CLB6	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
4	CLB5	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
3	CLB4	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
2	CLB3	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
1	CLB2	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
0	CLB1	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.41 SOFTPRES18 Register (Offset = 158h) [Reset = 0000000Xh]

SOFTPRES18 is shown in Figure 3-385 and described in Table 3-391.

Return to the Summary Table.

FSI Software Reset register

Figure 3-385. SOFTPRES18 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED				FSIRX_D	FSIRX_C	FSIRX_B	FSIRX_A
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				FSITX_D	FSITX_C	FSITX_B	FSITX_A
R/W-0h				R-Xh	R-Xh	R-Xh	R-Xh

Table 3-391. SOFTPRES18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	Reserved
19	FSIRX_D	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
18	FSIRX_C	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
17	FSIRX_B	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
16	FSIRX_A	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
15-4	RESERVED	R/W	0h	Reserved
3	FSITX_D	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
2	FSITX_C	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
1	FSITX_B	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
0	FSITX_A	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.42 SOFTPRES19 Register (Offset = 15Ch) [Reset = 00000000h]

SOFTPRES19 is shown in Figure 3-386 and described in Table 3-392.

Return to the Summary Table.

LIN Software Reset register

Figure 3-386. SOFTPRES19 Register

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

Table 3-392. SOFTPRES19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	LIN_B	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
0	LIN_A	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.43 SOFTPRES20 Register (Offset = 160h) [Reset = X0000000h]

SOFTPRES20 is shown in Figure 3-387 and described in Table 3-393.

Return to the Summary Table.

PMBUS Software Reset register

Figure 3-387. SOFTPRES20 Register

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

Table 3-393. SOFTPRES20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	Xh	Reserved
1	RESERVED	R	Xh	Reserved
0	PMBUS_A	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.44 SOFTPRES21 Register (Offset = 164h) [Reset = 00000000h]

SOFTPRES21 is shown in Figure 3-388 and described in Table 3-394.

Return to the Summary Table.

DCC Software Reset register

Figure 3-388. SOFTPRES21 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					DCC3	DCC2	DCC1
R-0-0h					R/W-0h	R/W-0h	R/W-0h

Table 3-394. SOFTPRES21 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	DCC3	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
1	DCC2	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
0	DCC1	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.45 SOFTPRES23 Register (Offset = 16Ch) [Reset = 00000001h]

SOFTPRES23 is shown in Figure 3-389 and described in Table 3-395.

Return to the Summary Table.

ETHERCAT Software Reset register

Figure 3-389. SOFTPRES23 Register

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

Table 3-395. SOFTPRES23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	ETHERCAT	R/W	1h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.46 SOFTPRES25 Register (Offset = 174h) [Reset = 00000000h]

SOFTPRES25 is shown in Figure 3-390 and described in Table 3-396.

Return to the Summary Table.

To be implemented only for CPU1

Figure 3-390. SOFTPRES25 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					HRCAL2	HRCAL1	HRCAL0
R-0-0h					R/W-0h	R/W-0h	R/W-0h

Table 3-396. SOFTPRES25 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	HRCAL2	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
1	HRCAL1	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
0	HRCAL0	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.47 SOFTPRES27 Register (Offset = 17Ch) [Reset = 00000000h]

SOFTPRES27 is shown in Figure 3-391 and described in Table 3-397.

Return to the Summary Table.

EPG Software Reset register

Figure 3-391. SOFTPRES27 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							EPG1
R-0-0h							R/W-0h

Table 3-397. SOFTPRES27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	EPG1	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.48 SOFTPRES28 Register (Offset = 180h) [Reset = 0000XX0Xh]

SOFTPRES28 is shown in Figure 3-392 and described in Table 3-398.

Return to the Summary Table.

ADCCHECKER Software Reset register

Figure 3-392. SOFTPRES28 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED		RESERVED	RESERVED	RESERVED	ADCSEAGGRCPU3	ADCSEAGGRCPU2	ADCSEAGGRCPU1
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED						ADCCHECKER10	ADCCHECKER9
R-Xh						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
ADCCHECKER8	ADCCHECKER7	ADCCHECKER6	ADCCHECKER5	ADCCHECKER4	ADCCHECKER3	ADCCHECKER2	ADCCHECKER1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-Xh	R-Xh

Table 3-398. SOFTPRES28 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	ADCSEAGGRCPU3	R/W	0h	ADC Safety Checker Error Aggregator Module for CPU 1 1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
17	ADCSEAGGRCPU2	R/W	0h	ADC Safety Checker Error Aggregator Module for CPU 2 1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
16	ADCSEAGGRCPU1	R/W	0h	ADC Safety Checker Error Aggregator Module for CPU 1 1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
15-10	RESERVED	R	Xh	Reserved
9	ADCCHECKER10	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
8	ADCCHECKER9	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
7	ADCCHECKER8	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

Table 3-398. SOFTPRES28 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	ADCCHECKER7	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
5	ADCCHECKER6	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
4	ADCCHECKER5	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
3	ADCCHECKER4	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
2	ADCCHECKER3	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
1	ADCCHECKER2	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
0	ADCCHECKER1	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.49 SOFTPRES30 Register (Offset = 188h) [Reset = 00000000h]

SOFTPRES30 is shown in Figure 3-393 and described in Table 3-399.

Return to the Summary Table.

SENT Software Reset register

Figure 3-393. SOFTPRES30 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		SENT6	SENT5	SENT4	SENT3	SENT2	SENT1
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-399. SOFTPRES30 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	SENT6	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
4	SENT5	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
3	SENT4	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
2	SENT3	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
1	SENT2	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
0	SENT1	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.50 SOFTPRES32 Register (Offset = 190h) [Reset = 00000000h]

SOFTPRES32 is shown in Figure 3-394 and described in Table 3-400.

Return to the Summary Table.

WADI Software Reset register

Figure 3-394. SOFTPRES32 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						WADI2	WADI1
R-0-0h						R/W-0h	R/W-0h

Table 3-400. SOFTPRES32 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	WADI2	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn
0	WADI1	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: CPUx.SYSRSn

3.13.8.51 PARITY_TEST_ALT1 Register (Offset = 1B8h) [Reset = 00000000h]

PARITY_TEST_ALT1 is shown in Figure 3-395 and described in Table 3-401.

Return to the Summary Table.

Enables parity test

Figure 3-395. PARITY_TEST_ALT1 Register

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

Table 3-401. PARITY_TEST_ALT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	TESTEN	R/W	0h	1010: Parity test feature is enabled All other values: Parity test feature is disabled Note: (1) When the parity test feature is enabled, actual registers are not accessible in the memory map. Instead, the parity values (final XOR output indicating the parity error) are accessible. Parity is computed for every byte and the corresponding parity error value is available at the bit-0 of every byte. Value of '1' written to the parity bit after enabling the parity test feature can be used to inject the error by inverting the stored parity value. (2) It is recommended to leave the field as 0101 or 0000 after completing the parity test. Reset type: SYSRSn

3.13.9 WD_REGS Registers

Table 3-402 lists the memory-mapped registers for the WD_REGS registers. All register offset addresses not listed in Table 3-402 should be considered as reserved locations and the register contents should not be modified.

Table 3-402. WD_REGS Registers

Offset	Acronym	Register Name	Protection
44h	SCSR	System Control & Status Register	
46h	WDCNTR	Watchdog Counter Register	
4Ah	WDKEY	Watchdog Reset Key Register	
50h	SYNCBUSYWD	SYNCBUSY status for Watchdog Register	
52h	WDCR	Watchdog Control Register	
54h	WDWCR	Watchdog Windowed Control Register	

Complex bit access types are encoded to fit into small table cells. Table 3-403 shows the codes that are used for access types in this section.

Table 3-403. WD_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.13.9.1 SCSR Register (Offset = 44h) [Reset = 0005h]

SCSR is shown in Figure 3-396 and described in Table 3-404.

Return to the Summary Table.

System Control & Status Register

Figure 3-396. SCSR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					WDINTS	WDENINT	WDOVERRIDE
R-0-0h					R-1h	R/W-0h	R/W1S-1h

Table 3-404. SCSR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-3	RESERVED	R-0	0h	Reserved
2	WDINTS	R	1h	Watchdog Interrupt Status This bit indicates the state of the active-low watchdog interrupt signal (synchronized to SYSCLK). If the watchdog interrupt is used to wake the system from a low-power mode, then that mode should only be entered while this bit is high. Likewise, this bit must go high before the watchdog can be safely disabled and re-enabled. 0h = The watchdog interrupt signal is active. 1h = The watchdog interrupt signal is inactive. Reset type: SYSRSn
1	WDENINT	R/W	0h	Watchdog Interrupt Enable/Reset Disable This bit determines whether the watchdog triggers an interrupt (WAKE/WDOG) or a reset (WDRS) when the counter expires. The error event is also generated to ESM on counter expiration 0h = Counter expiration triggers a reset. This is the default state on power-up and after any system reset. 1h = Counter expiration triggers an interrupt. Reset type: SYSRSn
0	WDOVERRIDE	R/W1S	1h	Watchdog Enable Lock Writing a 1 to this bit clears it and locks the WDDIS bit in the WDCR register. The bit will remain in this state until the next system reset. Reads of this bit return its current value. Writing a 0 to this bit has no effect. Reset type: SYSRSn

3.13.9.2 WDCNTR Register (Offset = 46h) [Reset = 0000h]

WDCNTR is shown in Figure 3-397 and described in Table 3-405.

Return to the Summary Table.

Watchdog Counter Register

Figure 3-397. WDCNTR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
WDCNTR							
R-0h							

Table 3-405. WDCNTR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R-0	0h	Reserved
7-0	WDCNTR	R	0h	Watchdog Counter These bits contain the current value of the watchdog counter. This counter increments with each WDCLK (INTOSC1) cycle. If the counter overflows, either an interrupt or a reset is generated based on the value of the WDINTEN bit in the SCSR register. If the correct value is written to the WDKEY register, this counter is reset to zero. Note : Reads to this register before watchdog is enabled will return unpredictable value Reset type: IORSn

3.13.9.3 WDKEY Register (Offset = 4Ah) [Reset = 0000h]

WDKEY is shown in Figure 3-398 and described in Table 3-406.

Return to the Summary Table.

Watchdog Reset Key Register

Figure 3-398. WDKEY Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
WDKEY							
R/W-0h							

Table 3-406. WDKEY Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R-0	0h	Reserved
7-0	WDKEY	R/W	0h	Watchdog Counter Reset Writing 0x55 followed by 0xAA will cause the watchdog counter to reset to zero, preventing an overflow. Writing other values has no effect. Reads of this register return the value of the WDCR register. Reset type: IORSn

3.13.9.4 SYNCBUSYWD Register (Offset = 50h) [Reset = 0000h]

SYNCBUSYWD is shown in Figure 3-399 and described in Table 3-407.

Return to the Summary Table.

SYNCBUSY status for Watchdog Register

Figure 3-399. SYNCBUSYWD Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						WDCR	BUSY
R-0-0h						R-0h	R-0h

Table 3-407. SYNCBUSYWD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-2	RESERVED	R-0	0h	Reserved
1	WDCR	R	0h	This status bit indicates write to the register is in progress 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn
0	BUSY	R	0h	This status bit indicates write to any of the following registers (OR_REDUCE) is in progress or not. WDCR 0 : Not BUSY - No synchronization in progress 1 : BUSY - Synchronization is in progress Reset type: SYSRSn

3.13.9.5 WDCR Register (Offset = 52h) [Reset = 0000h]

WDCR is shown in Figure 3-400 and described in Table 3-408.

Return to the Summary Table.

Watchdog Control Register
This memory mapped register requires a delay of 69 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 69 NOP instructions.

Figure 3-400. WDCR Register

15	14	13	12	11	10	9	8
RESERVED				WDPRECLKDIV			
R-0-0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED	WDDIS	WDCHK			WDPS		
R/W1S-0h	R/W-0h	R-0/W-0h			R/W-0h		

Table 3-408. WDCR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R-0	0h	Reserved
11-8	WDPRECLKDIV	R/W	0h	Watchdog Clock Pre-divider These bits determine the watchdog clock pre-divider, which is the first of the two dividers between INTOSC1 and the watchdog counter clock (WDCLK). The frequency of WDCLK is given by the formulas: PREDIVCLK = INTOSC1 / Pre-divider WDCLK = PREDIVCLK / Prescaler The watchdog reset or interrupt pulse is 512 INTOSC1 cycles long, so the counter period must be longer. To guarantee this, the product of the prescaler and pre-divider must be greater than or equal to four. The default pre-divider value is 512. Reset type: IORSn
7	RESERVED	R/W1S	0h	Reserved
6	WDDIS	R/W	0h	Watchdog Disable Setting this bit disables the watchdog module. Clearing this bit enables the watchdog module. This bit can be locked by the WDOVERRIDE bit in the SCSR register. The watchdog is enabled on reset. Reset type: IORSn
5-3	WDCHK	R-0/W	0h	Watchdog Check Bits During any write to this register, these bits must be written with the value 101 (binary). Writing any other value will immediately trigger the watchdog reset or interrupt. Reset type: IORSn
2-0	WDPS	R/W	0h	Watchdog Clock Prescaler These bits determine the watchdog clock prescaler, which is the second of the two dividers between INTOSC1 and the watchdog counter clock (WDCLK). The frequency of WDCLK is given by the formulas: PREDIVCLK = INTOSC1 / Pre-divider WDCLK = PREDIVCLK / Prescaler The watchdog reset or interrupt pulse is 512 INTOSC1 cycles long, so the counter period must be longer. To guarantee this, the product of the prescaler and pre-divider must be greater than or equal to four. The default prescaler value is 1. Reset type: IORSn

3.13.9.6 WDWCR Register (Offset = 54h) [Reset = 0000h]

WDWCR is shown in Figure 3-401 and described in Table 3-409.

Return to the Summary Table.

Watchdog Windowed Control Register

Figure 3-401. WDWCR Register

15	14	13	12	11	10	9	8
RESERVED							RESERVED
R-0-0h							R-0h
7	6	5	4	3	2	1	0
MIN							
R/W-0h							

Table 3-409. WDWCR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8	RESERVED	R	0h	Reserved
7-0	MIN	R/W	0h	Watchdog Window Threshold These bits specify the lower limit of the watchdog counter reset window. If the counter is reset via the WDKEY register before the counter value reaches the value in this register, the watchdog immediately triggers a reset or interrupt. Reset type: IORSn

3.13.10 CPUTIMER_REGS Registers

Table 3-410 lists the memory-mapped registers for the CPUTIMER_REGS registers. All register offset addresses not listed in Table 3-410 should be considered as reserved locations and the register contents should not be modified.

Table 3-410. CPUTIMER_REGS Registers

Offset	Acronym	Register Name	Protection
0h	TIM	CPU-Timer, Counter Register	
4h	PRD	CPU-Timer, Period Register	
8h	TCR	CPU-Timer, Control Register	
Ch	TPR	CPU-Timer, Prescale Register	
Eh	TPRH	CPU-Timer, Prescale Register High	

Complex bit access types are encoded to fit into small table cells. Table 3-411 shows the codes that are used for access types in this section.

Table 3-411. CPUTIMER_REGS Access Type Codes

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

3.13.10.1 TIM Register (Offset = 0h) [Reset = 0000FFFFh]

TIM is shown in Figure 3-402 and described in Table 3-412.

Return to the Summary Table.

CPU-Timer, Counter Register

Figure 3-402. TIM Register

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

Table 3-412. TIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MSW	R/W	0h	CPU-Timer Counter Registers The TIMH register holds the high 16 bits of the current 32-bit count of the timer. The TIMH:TIM decrements by one every (TDDRH:TDDR+1) clock cycles, where TDDRH:TDDR is the timer prescale dividedown value. When the TIMH:TIM decrements to zero, the TIMH:TIM register is reloaded with the period value contained in the PRDH:PRD registers. The timer interrupt (TINT) signal is generated. Reset type: SYSRSn
15-0	LSW	R/W	FFFFh	CPU-Timer Counter Registers The TIM register holds the low 16 bits of the current 32-bit count of the timer. The TIMH:TIM decrements by one every (TDDRH:TDDR+1) clock cycles, where TDDRH:TDDR is the timer prescale dividedown value. When the TIMH:TIM decrements to zero, the TIMH:TIM register is reloaded with the period value contained in the PRDH:PRD registers. The timer interrupt (TINT) signal is generated. Reset type: SYSRSn

3.13.10.2 PRD Register (Offset = 4h) [Reset = 0000FFFFh]

PRD is shown in Figure 3-403 and described in Table 3-413.

Return to the Summary Table.

CPU-Timer, Period Register

Figure 3-403. PRD Register

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

Table 3-413. PRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MSW	R/W	0h	CPU-Timer Period Registers The PRDH register holds the high 16 bits of the 32-bit period. When the TIMH:TIM decrements to zero, the TIMH:TIM register is reloaded with the period value contained in the PRDH:PRD registers, at the start of the next timer input clock cycle (the output of the prescaler). The PRDH:PRD contents are also loaded into the TIMH:TIM when you set the timer reload bit (TRB) in the Timer Control Register (TCR). Reset type: SYSRSn
15-0	LSW	R/W	FFFFh	CPU-Timer Period Registers The PRD register holds the low 16 bits of the 32-bit period. When the TIMH:TIM decrements to zero, the TIMH:TIM register is reloaded with the period value contained in the PRDH:PRD registers, at the start of the next timer input clock cycle (the output of the prescaler). The PRDH:PRD contents are also loaded into the TIMH:TIM when you set the timer reload bit (TRB) in the Timer Control Register (TCR). Reset type: SYSRSn

3.13.10.3 TCR Register (Offset = 8h) [Reset = 0001h]

TCR is shown in Figure 3-404 and described in Table 3-414.

Return to the Summary Table.

CPU-Timer, Control Register

Figure 3-404. TCR Register

15	14	13	12	11	10	9	8
TIF	TIE	RESERVED		FREE	SOFT	RESERVED	
R/W1C-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	R-0h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		TRB	TSS	RESERVED			
R-0h		R/W-0h	R/W-0h	R-1h			

Table 3-414. TCR Register Field Descriptions

Bit	Field	Type	Reset	Description
15	TIF	R/W1C	0h	CPU-Timer Overflow Flag. TIF indicates whether a timer overflow has happened since TIF was last cleared. TIF is not cleared automatically and does not need to be cleared to enable the next timer interrupt. Reset type: SYSRSn 0h (R/W) = The CPU-Timer has not decremented to zero. Writes of 0 are ignored. 1h (R/W) = This flag gets set when the CPU-timer decrements to zero. Writing a 1 to this bit clears the flag.
14	TIE	R/W	0h	CPU-Timer Interrupt Enable. Reset type: SYSRSn 0h (R/W) = The CPU-Timer interrupt is disabled. 1h (R/W) = The CPU-Timer interrupt is enabled. If the timer decrements to zero, and TIE is set, the timer asserts its interrupt request.
13-12	RESERVED	R	0h	Reserved
11	FREE	R/W	0h	If the FREE bit is set to 1, then, upon a software breakpoint, the timer continues to run. If FREE is 0, then the SOFT bit controls the emulation behavior. Reset type: SYSRSn 0h (R/W) = Stop after the next decrement of the TIMH:TIM (hard stop) (SOFT bit controls the emulation behavior) 1h (R/W) = Free Run (SOFT bit is don't care, counter is free running)
10	SOFT	R/W	0h	If the FREE bit is set to 1, then, upon a software breakpoint, the timer continues to run (that is, free runs). In this case, SOFT is a don't care. But if FREE is 0, then SOFT takes effect. Reset type: SYSRSn 0h (R/W) = Stop after the next decrement of the TIMH:TIM (hard stop). (ONLY if FREE=0, if FREE=1 this bit is don't care) 1h (R/W) = Stop after the TIMH:TIM decrements to 0 (soft stop) In the SOFT STOP mode, the timer generates an interrupt before shutting down (since reaching 0 is the interrupt causing condition). (ONLY if FREE=0, if FREE=1 this bit is don't care)
9-6	RESERVED	R	0h	Reserved

Table 3-414. TCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	TRB	R/W	0h	Timer reload Reset type: SYSRSn 0h (R/W) = The TRB bit is always read as zero. Writes of 0 are ignored. 1h (R/W) = When you write a 1 to TRB, the TIMH:TIM is loaded with the value in the PRDH:PRD, and the prescaler counter (PSCH:PSC) is loaded with the value in the timer dividedown register (TDDRH:TDDR).
4	TSS	R/W	0h	CPU-Timer stop status bit. TSS is a 1-bit flag that stops or starts the CPU-timer. Reset type: SYSRSn 0h (R/W) = Reads of 0 indicate the CPU-timer is running. To start or restart the CPU-timer, set TSS to 0. At reset, TSS is cleared to 0 and the CPU-timer immediately starts. 1h (R/W) = Reads of 1 indicate that the CPU-timer is stopped. To stop the CPU-timer, set TSS to 1.
3-0	RESERVED	R	1h	Reserved

3.13.10.4 TPR Register (Offset = Ch) [Reset = 0000h]

TPR is shown in Figure 3-405 and described in Table 3-415.

Return to the Summary Table.

CPU-Timer, Prescale Register

Figure 3-405. TPR Register

15	14	13	12	11	10	9	8
PSC							
R-0h							
7	6	5	4	3	2	1	0
TDDR							
R/W-0h							

Table 3-415. TPR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	PSC	R	0h	CPU-Timer Prescale Counter. These bits hold the current prescale count for the timer. For every timer clock source cycle that the PSCH:PSC value is greater than 0, the PSCH:PSC decrements by one. One timer clock (output of the timer prescaler) cycle after the PSCH:PSC reaches 0, the PSCH:PSC is loaded with the contents of the TDDRH:TDDR, and the timer counter register (TIMH:TIM) decrements by one. The PSCH:PSC is also reloaded whenever the timer reload bit (TRB) is set by software. The PSCH:PSC can be checked by reading the register, but it cannot be set directly. It must get its value from the timer divide-down register (TDDRH:TDDR). At reset, the PSCH:PSC is set to 0. Reset type: SYSRSn
7-0	TDDR	R/W	0h	CPU-Timer Divide-Down. Every (TDDRH:TDDR + 1) timer clock source cycles, the timer counter register (TIMH:TIM) decrements by one. At reset, the TDDRH:TDDR bits are cleared to 0. To increase the overall timer count by an integer factor, write this factor minus one to the TDDRH:TDDR bits. When the prescaler counter (PSCH:PSC) value is 0, one timer clock source cycle later, the contents of the TDDRH:TDDR reload the PSCH:PSC, and the TIMH:TIM decrements by one. TDDRH:TDDR also reloads the PSCH:PSC whenever the timer reload bit (TRB) is set by software. Reset type: SYSRSn

3.13.10.5 TPRH Register (Offset = Eh) [Reset = 0000h]

TPRH is shown in Figure 3-406 and described in Table 3-416.

Return to the Summary Table.

CPU-Timer, Prescale Register High

Figure 3-406. TPRH Register

15	14	13	12	11	10	9	8
PSCH							
R-0h							
7	6	5	4	3	2	1	0
TDDRH							
R/W-0h							

Table 3-416. TPRH Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	PSCH	R	0h	See description of TIMERxTPR. Reset type: SYSRSn
7-0	TDDRH	R/W	0h	See description of TIMERxTPR. Reset type: SYSRSn

3.13.11 XINT_REGS Registers

Table 3-417 lists the memory-mapped registers for the XINT_REGS registers. All register offset addresses not listed in Table 3-417 should be considered as reserved locations and the register contents should not be modified.

Table 3-417. XINT_REGS Registers

Offset	Acronym	Register Name	Protection
0h	XINT1CR	XINT1 configuration register	
2h	XINT2CR	XINT2 configuration register	
4h	XINT3CR	XINT3 configuration register	
6h	XINT4CR	XINT4 configuration register	
8h	XINT5CR	XINT5 configuration register	
10h	XINT1CTR	XINT1 counter register	
12h	XINT2CTR	XINT2 counter register	
14h	XINT3CTR	XINT3 counter register	

Complex bit access types are encoded to fit into small table cells. Table 3-418 shows the codes that are used for access types in this section.

Table 3-418. XINT_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.13.11.1 XINT1CR Register (Offset = 0h) [Reset = 0000h]

XINT1CR is shown in Figure 3-407 and described in Table 3-419.

Return to the Summary Table.

XINT1 configuration register

Figure 3-407. XINT1CR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				POLARITY		RESERVED	ENABLE
R-0-0h				R/W-0h		R-0-0h	R/W-0h

Table 3-419. XINT1CR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3-2	POLARITY	R/W	0h	00: Interrupt is selected as negative edge triggered 01: Interrupt is selected as positive edge triggered 10: Interrupt is selected as negative edge triggered 11: Interrupt is selected as positive or negative edge triggered Reset type: SYSRSn
1	RESERVED	R-0	0h	Reserved
0	ENABLE	R/W	0h	0: Interrupt Disabled 1: Interrupt Enabled Reset type: SYSRSn

3.13.11.2 XINT2CR Register (Offset = 2h) [Reset = 0000h]

XINT2CR is shown in Figure 3-408 and described in Table 3-420.

Return to the Summary Table.

XINT2 configuration register

Figure 3-408. XINT2CR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				POLARITY		RESERVED	ENABLE
R-0-0h				R/W-0h		R-0-0h	R/W-0h

Table 3-420. XINT2CR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3-2	POLARITY	R/W	0h	00: Interrupt is selected as negative edge triggered 01: Interrupt is selected as positive edge triggered 10: Interrupt is selected as negative edge triggered 11: Interrupt is selected as positive or negative edge triggered Reset type: SYSRSn
1	RESERVED	R-0	0h	Reserved
0	ENABLE	R/W	0h	0: Interrupt Disabled 1: Interrupt Enabled Reset type: SYSRSn

3.13.11.3 XINT3CR Register (Offset = 4h) [Reset = 0000h]

XINT3CR is shown in Figure 3-409 and described in Table 3-421.

Return to the Summary Table.

XINT3 configuration register

Figure 3-409. XINT3CR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				POLARITY		RESERVED	ENABLE
R-0-0h				R/W-0h		R-0-0h	R/W-0h

Table 3-421. XINT3CR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3-2	POLARITY	R/W	0h	00: Interrupt is selected as negative edge triggered 01: Interrupt is selected as positive edge triggered 10: Interrupt is selected as negative edge triggered 11: Interrupt is selected as positive or negative edge triggered Reset type: SYSRSn
1	RESERVED	R-0	0h	Reserved
0	ENABLE	R/W	0h	0: Interrupt Disabled 1: Interrupt Enabled Reset type: SYSRSn

3.13.11.4 XINT4CR Register (Offset = 6h) [Reset = 0000h]

XINT4CR is shown in Figure 3-410 and described in Table 3-422.

Return to the Summary Table.

XINT4 configuration register

Figure 3-410. XINT4CR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				POLARITY		RESERVED	ENABLE
R-0-0h				R/W-0h		R-0-0h	R/W-0h

Table 3-422. XINT4CR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3-2	POLARITY	R/W	0h	00: Interrupt is selected as negative edge triggered 01: Interrupt is selected as positive edge triggered 10: Interrupt is selected as negative edge triggered 11: Interrupt is selected as positive or negative edge triggered Reset type: SYSRSn
1	RESERVED	R-0	0h	Reserved
0	ENABLE	R/W	0h	0: Interrupt Disabled 1: Interrupt Enabled Reset type: SYSRSn

3.13.11.5 XINT5CR Register (Offset = 8h) [Reset = 0000h]

XINT5CR is shown in Figure 3-411 and described in Table 3-423.

Return to the Summary Table.

XINT5 configuration register

Figure 3-411. XINT5CR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				POLARITY		RESERVED	ENABLE
R-0-0h				R/W-0h		R-0-0h	R/W-0h

Table 3-423. XINT5CR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3-2	POLARITY	R/W	0h	00: Interrupt is selected as negative edge triggered 01: Interrupt is selected as positive edge triggered 10: Interrupt is selected as negative edge triggered 11: Interrupt is selected as positive or negative edge triggered Reset type: SYSRSn
1	RESERVED	R-0	0h	Reserved
0	ENABLE	R/W	0h	0: Interrupt Disabled 1: Interrupt Enabled Reset type: SYSRSn

3.13.11.6 XINT1CTR Register (Offset = 10h) [Reset = 0000h]

XINT1CTR is shown in Figure 3-412 and described in Table 3-424.

Return to the Summary Table.

XINT1 counter register

Figure 3-412. XINT1CTR Register

15	14	13	12	11	10	9	8
INTCTR							
R-0h							
7	6	5	4	3	2	1	0
INTCTR							
R-0h							

Table 3-424. XINT1CTR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	INTCTR	R	0h	This is a free running 16-bit up-counter that is clocked at the SYSCLKOUT rate. The counter value is reset to 0x0000 when a valid interrupt edge is detected and then continues counting until the next valid interrupt edge is detected. The counter must only be reset by the selected POLARITY edge as selected in the respective interrupt control register. When the interrupt is disabled, the counter will stop. The counter is a free-running counter and will wrap around to zero when the max value is reached. The counter is a read only register and can only be reset to zero by a valid interrupt edge or by reset. Reset type: SYSRSn

3.13.11.7 XINT2CTR Register (Offset = 12h) [Reset = 0000h]

XINT2CTR is shown in Figure 3-413 and described in Table 3-425.

Return to the Summary Table.

XINT2 counter register

Figure 3-413. XINT2CTR Register

15	14	13	12	11	10	9	8
INTCTR							
R-0h							
7	6	5	4	3	2	1	0
INTCTR							
R-0h							

Table 3-425. XINT2CTR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	INTCTR	R	0h	This is a free running 16-bit up-counter that is clocked at the SYSCLKOUT rate. The counter value is reset to 0x0000 when a valid interrupt edge is detected and then continues counting until the next valid interrupt edge is detected. The counter must only be reset by the selected POLARITY edge as selected in the respective interrupt control register. When the interrupt is disabled, the counter will stop. The counter is a free-running counter and will wrap around to zero when the max value is reached. The counter is a read only register and can only be reset to zero by a valid interrupt edge or by reset. Reset type: SYSRSn

3.13.11.8 XINT3CTR Register (Offset = 14h) [Reset = 0000h]

XINT3CTR is shown in Figure 3-414 and described in Table 3-426.

Return to the Summary Table.

XINT3 counter register

Figure 3-414. XINT3CTR Register

15	14	13	12	11	10	9	8
INTCTR							
R-0h							
7	6	5	4	3	2	1	0
INTCTR							
R-0h							

Table 3-426. XINT3CTR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	INTCTR	R	0h	This is a free running 16-bit up-counter that is clocked at the SYSCLKOUT rate. The counter value is reset to 0x0000 when a valid interrupt edge is detected and then continues counting until the next valid interrupt edge is detected. The counter must only be reset by the selected POLARITY edge as selected in the respective interrupt control register. When the interrupt is disabled, the counter will stop. The counter is a free-running counter and will wrap around to zero when the max value is reached. The counter is a read only register and can only be reset to zero by a valid interrupt edge or by reset. Reset type: SYSRSn

Chapter 4

ROM Code and Peripheral Booting

This chapter explains the boot procedure, the available boot modes, and the various details of the ROM code including memory maps, initializations, reset handling, and status information.

4.1 Introduction

The purpose of this chapter is to explain the boot read-only memory (ROM) code functionality for the C29x CPU core's, including the boot procedure. This chapter also discusses the functions and features of the boot ROM code, and provides details about the ROM memory-map contents. On every reset, the device executes a boot sequence in the ROM depending on the reset type and boot configuration. This sequence initializes the device to run the application code. For the CPU, the boot ROM also contains peripheral bootloaders that can be used to load an application into RAM. These bootloaders can be disabled for safety or security purposes.

See Table 4-1 for details on available boot features for the C29x CPU. Additionally, Table 4-2 shows the sizes of the various ROMs on the device.

Various tables are provided in ROM for use in software library, refer to Section 4.7.5 for more details.

Table 4-1. Boot System Overview

Boot Feature	CPU
Initial boot process	Device reset
Boot mode selection	GPIOs
Boot modes supported	Flash boot RAM boot Wait boot Parallel IO CAN CAN-FD I2C SPI UART

Table 4-2. ROM Memory

ROM	Size
CPU1 boot ROM	128KB
CPU2 boot ROM	32KB
CPU3 boot ROM	32KB

CPU2 and CPU3 doesn't have any BootROM code and just contains default NMI handler. During initialization application needs to re-assign the handler to override the default handler. CPU2/3 default NMI handlers points to infinite while loop. This section further explains in detail on the CPU1 Boot ROM flow.

4.1.1 ROM Related Collateral

Foundational Materials

- Bootloading 101 (Video)

Expert Materials

- *C2000 Software Controlled Firmware Update Process Application Report*

4.2 Device Boot Sequence

Table 4-3 describes the general boot ROM procedure each time the CPU1 core is reset.

During boot, boot ROM code updates a boot status location in RAM that details the actions taken during this process. Refer to Section 4.7.9 for more details.

Table 4-3. CPU1 Boot ROM Sequence

Step	CPU1 Action
1	Flash Read Interface (FRI) wait state configuration
2	Enable Watchdog
3	Zone0 full debug password configured from OTP into SSU registers
4	UID (Unique ID) is configured from OTP into SSU registers
5	On PORESETn only, All CPU RAMs (LPAx,LDAx, CPAx and CDAx) are initialized
6	Critical Trims (APLL, PMM, OSC, Flash) are loaded from OTP and device configuration registers are programmed
7	ESM configurations are performed for Group0 events
8	SIC (Safe Interconnect) is enabled
9	UPP (User Protection Policy) revision from User-OTP is configured into SSU register
10	Error status pin configuration input from User-OTP is configured
11	External crystal power-up if enabled in User-OTP
12	Reading the Device Configurations from OTP into DCx Registers
13	Load non-critical (ADC, DAC) trims
14	SSU configurations based on User-OTP inputs which include : 1. SSU register self-test 2. SSU register configurations
15	Lock DCx (Device Configuration), PARTID, MCUCNF26 and PERxSYSCONFIG (Peripheral System Configuration) registers
16	Wait for RAM initializaion, done only on PORESETn
17	Clear PORRESETn and XRSn reset cause on PORESETn and only clear XRSn reset cause on XRSn
18	Pull-ups are enabled on unbonded IOs
19	The boot mode GPIO pins are polled to determine the boot mode to run. Boot loader is executed based on boot mode/configurations. Refer to Section 4.5.2 for a flow chart of the boot sequences.
20	RAMOPEN for LINK1 which includes: LPA0 and LDA0-7
21	Lock and Commit LINK1 RAMPOPEN by writing to SSU registers based on User-OTP inputs
22	APR's (Access Protection Regions) are set from User-OTP configurations
23	Disable watchdog for Link1 bootloaders execution
24	Bootloader process under Link1 execution
25	Clear Link1 RAMOPEN
26	Jump to C29 Application Link2

4.3 Device Boot Modes

This section explains the default boot modes, as well as all the available custom boot modes supported on this device. The boot ROM uses the boot mode select, general purpose input/output (GPIO) pins to determine the boot mode configuration.

4.3.1 Default Boot Modes

Table 4-4 shows the boot mode options available for selection by the default boot mode select pins. Users have the option to program the device to customize the boot modes selectable in the boot-up table as well as the boot mode select pin GPIOs used. Default BMSP (Boot Mode Select Pin) used are GPIO72 (BMSP1) and GPIO84 (BMSP0).

Table 4-4. Device Default Boot Modes

Boot Mode	GPIO72 (Default boot mode select pin 1)	GPIO84 (Default boot mode select pin 0)
Parallel IO	0	0
UART	0	1
CAN	1	0
Flash	1	1

Refer to Section 4.4 for details of boot configurations.

Refer to Section 4.7.6.2 for details of the boot modes that use a peripheral boot loader.

Refer to Section 4.7.7 for GPIOs used for selecting the boot modes.

Note

All the peripheral boot modes that are supported use the first instance of the peripheral module (SPIA, I2CA, CANA, and so forth). Whenever these boot modes are referred to in this chapter, such as SPI boot, the mode is actually referring to the first module instance, which means the SPI boot on the SPIA port. The same applies to the other peripheral boot modes.

4.3.2 Custom Boot Modes

Once the user programs a custom boot table in user OTP, an entry in the custom table is used for booting. Users can customize the boot mode select pins in the end system design by programming the BOOTPIN_CONFIG location in user OTP. This allows customers to use 0, 1, 2, or 3 boot mode select pins as needed. You can also customize the boot definition table and indicate which location to boot from by programming the boot mode definition table in the BOOTDEF location of user OTP. Table 4-5 show the options for various boot modes.

Note

All peripheral boot modes supported in Table 4-5 use the first instance of the peripheral modules (that is SPIA, I2CA, and so on).

Table 4-5. CPU1 Boot Modes

Boot Mode Number	Boot Modes
0	Parallel
1	UART
2	CAN
3	Flash
4	Wait
5	RAM
6	SPI
7	I2C
8	CAN-FD

4.4 Device Boot Configurations

This section details what boot configurations are available and how to configure them. This device supports from zero boot mode select pins up to three boot mode select pins and from one configured boot mode up to eight configured boot modes.

To change and configure the device from the default settings to custom settings for your application, use the following process:

1. Determine all the various ways you want application to be able to boot. (For example: Primary boot option of Flash boot for your main application, secondary boot option of CAN boot for firmware updates, tertiary boot option of SPI boot for debugging, and so on.)
2. Based on the number of boot modes needed, determine how many boot mode select pins (BMSPs) are required to select between your selected boot modes. (For example: Two BMSPs are required to select between three boot mode options.)
3. Assign the required BMSPs to a physical GPIO pin. (For example, BMSP0 to GPIO10, BMSP1 to GPIO51, and BMSP2 left as default that is disabled.). Refer to Section 4.4.1 for all the details on performing these configurations.
4. Assign the determined boot mode definitions to indexes in your custom boot table that correlate to the decoded value of the BMSPs. For example, BOOTDEF0 = Boot to Flash, BOOTDEF1 = CAN Boot, BOOTDEF2 = SPI Boot; all other BOOTDEFx remain as default/nothing). Refer to Section 4.4.2 for all the details on setting up and configuring the custom boot mode table.

Additionally, Section 4.4.3 provides some example use cases on how to configure the boot mode select pins and custom boot tables.

4.4.1 Configuring Boot Mode Pins

This section explains how the boot mode select pins are customized by the user, by programming the BOOTPIN_CONFIG location (refer to Table 4-6), in the user-configurable OTP. The location in the OTP is BOOTPIN-CONFIG. When debugging, EMU_BOOTPIN_CONFIG register in SSU_GEN_REGS is the emulation equivalent of BOOTPIN_CONFIG, and can be programmed to experiment with different boot modes without writing to OTP. The device can be programmed to use **zero**, **one**, **two**, or **three** boot mode select pins as needed.

BMSP configuration and boot definition table is either read from User-OTP or SSU registers based on debugger connection status as explained below :

- If debugger is connected then emulation boot flow is followed, where following SSU registers are used to determine GPIO to be used :
  - EMU_BOOTPIN_CONFIG
  - EMU_BOOTDEF_LOW
  - EMU_BOOTDEF_HIGH
- If debugger is not connected then following User-OTP locations are used to determine the bootmodes :
  - BOOTPIN_CONFIG
  - BOOTDEF_LOW
  - BOOTDEF_HIGH

Table 4-6. BOOTPIN-CONFIG Bit Fields

Bit	Name	Description
31:24	Key	Write 0x5A to these 8-bits to tell the boot ROM code that the bits in this register are valid.
23:16	Boot Mode Select Pin 2 (BMSP2)	Refer to BMSP0 description.
15:8	Boot Mode Select Pin 1 (BMSP1)	Refer to BMSP0 description.
7:0	Boot Mode Select Pin 0 (BMSP0)	Set to the GPIO pin to be used during boot (GPIO0 up to GPIO254). 0x0 = GPIO0 0x01 = GPIO1, and so on. Writing 0xFF disables this BMSP and this pin is no longer used to select the boot mode.

Note

GPIO that can be either digital and analog type pins, digital type inputs are possible on these pins provided the software writes to the GPIOHAMSEL register bits.

The following GPIOs that are not available on any package cannot be used as a boot mode select pin. If selected for a particular BMSP, the boot ROM automatically selects the factory default GPIOs for BMSP0 and BMSP1. Factory default for BMSP2 is 0xFF, which disables the BMSP.

Table 4-7. Standalone Boot Mode Select Pin Decoding

BOOTPIN_CONFIG Key	BMSP0	BMSP1	BMSP2	Realized Boot Mode
!= 0x5A	Don't Care	Don't Care	Don't Care	Boot as defined by the factory default BMSPs.
= 0x5A	0xFF	0xFF	0xFF	Boot as defined in the boot table for boot mode 0 (All BMSPs disabled).
	Valid GPIO	0xFF	0xFF	Boot as defined by the value of BMSP0 (BMSP1 and BMSP2 disabled).
	0xFF	Valid GPIO	0xFF	Boot as defined by the value of BMSP1 (BMSP0 and BMSP2 disabled).
	0xFF	0xFF	Valid GPIO	Boot as defined by the value of BMSP2 (BMSP0 and BMSP1 disabled)
	Valid GPIO	Valid GPIO	0xFF	Boot as defined by the values of BMSP0 and BMSP1 (BMSP2 disabled).
	Valid GPIO	0xFF	Valid GPIO	Boot as defined by the values of BMSP0 and BMSP2 (BMSP1 disabled).
	0xFF	Valid GPIO	Valid GPIO	Boot as defined by the values of BMSP1 and BMSP2 (BMSP0 disabled).
	Valid GPIO	Valid GPIO	Valid GPIO	Boot as defined by the values of BMSP0, BMSP1, and BMSP2.
	Invalid GPIO	Valid GPIO	Valid GPIO	BMSP0 is reset to the factory default BMSP0 GPIO. Boot as defined by the values of BMSP0, BMSP1, and BMSP2.
	Valid GPIO	Invalid GPIO	Valid GPIO	BMSP1 is reset to the factory default BMSP1 GPIO. Boot as defined by the values of BMSP0, BMSP1, and BMSP2.
	Valid GPIO	Valid GPIO	Invalid GPIO	BMSP2 is reset to the factory default state, which is disabled. Boot as defined by the values of BMSP0 and BMSP1.

Note

When decoding the boot mode, BMSP0 is the least-significant bit and BMSP2 is the most-significant bit of the boot table index value. It is recommended when disabling BMSPs to start with disabling BMSP2. For example, in an instance when only using BMSP2 (BMSP1 and BMSP0 are disabled), then only the boot table indexes of 0 and 4 are selectable. In the instance when using only BMSP0, then the selectable boot table indexes are 0 and 1.

4.4.2 Configuring Boot Mode Table Options

This section explains how to configure the boot definition table, BOOTDEF, for the device and the associated boot options (refer to Table 4-8). The 64-bit location is located in user-configurable OTP in the BOOTDEF_LOW and BOOTDEF_HIGH locations. When debugging, EMU_BOOTDEF_LOW and EMU_BOOTDEF_HIGH are the emulation equivalents of BOOTDEF_LOW and BOOTDEF_HIGH, and can be programmed to experiment with different boot mode options without writing to OTP.

The range of customization to the boot definition table depends on how many boot mode select pins (BMSP) are being used. For example, 0 BMSPs equals to 1 table entry, 1 BMSP equals to 2 table entries, 2 BMSPs equals to 4 table entries, and 3 BMSPs equals to 8 table entries. Refer to Section 4.4.3 for examples on how to setup the BOOTPIN_CONFIG and BOOTDEF values.

Table 4-8. BOOTDEF Bit Fields

BOOTDEF Name	Byte Position	Name	Description
BOOT_DEF0	7:0	[3:0] BOOT_DEF0 Mode	Set the boot mode number from Section 4.3.2. Any unsupported boot mode causes the device to either go to wait boot (debugger connected) or boot to Flash (standalone).
		[7:4] BOOT_DEF0 Options	Set alternate/additional boot options. This can include changing the GPIOs for a particular boot peripheral or specifying a different Flash entry point. Refer to Section 4.7.7 for valid BOOTDEF values to set in the table.
BOOT_DEF1	15:8	BOOT_DEF1 Mode/Options	Refer to BOOT_DEF0 description.
BOOT_DEF2	23:16	BOOT_DEF2 Mode/Options	
BOOT_DEF3	31:24	BOOT_DEF3 Mode/Options	
BOOT_DEF4	39:32	BOOT_DEF4 Mode/Options	
BOOT_DEF5	47:40	BOOT_DEF5 Mode/Options	
BOOT_DEF6	55:48	BOOT_DEF6 Mode/Options	
BOOT_DEF7	63:56	BOOT_DEF7 Mode/Options	

4.4.3 Boot Mode Example Use Cases

This section demonstrates some use cases for configuring the boot mode select pins and boot modes.

4.4.3.1 Zero Boot Mode Select Pins

This use-case demonstrates a scenario for an application that does not use any boot mode select pins and always has the device boot to Flash.

1. Program the BOOTPIN_CONFIG location in OTP as follows:
   - Set BOOTPIN_CONFIG.BMSP0 to 0xFF
   - Set BOOTPIN_CONFIG.BMSP1 to 0xFF
   - Set BOOTPIN_CONFIG.BMSP2 to 0xFF
   - Set BOOTPIN_CONFIG.KEY to 0x5A for boot ROM to treat these register bits as valid and use the custom boot table.
2. Program the BOOTDEF location options for the device. This essentially sets up a device-specific boot mode table. Refer to Section 4.7.7 for valid BOOTDEF values to set in the table.
   - Set BOOTDEF.BOOTDEF0 to 0x03 for booting to Flash (entry address option 0). This sets Flash boot to boot table index 0.
   - Refer to Section 4.7.2 for the available Flash entry points.

Table 4-9. Zero Boot Pin Boot Table Result

Boot Mode Table Number	Boot Mode
0	Flash Boot (0x03)

4.4.3.2 One Boot Mode Select Pin

This use-case demonstrates a scenario for an application using one boot mode select pin to select between booting to Flash or using CAN boot.

1. Program the BOOTPIN_CONFIG location in OTP as follows:
   - Set BOOTPIN_CONFIG.BMSP0 to a user specified GPIO, such as 0x0 for GPIO0
   - Set BOOTPIN_CONFIG.BMSP1 to 0xFF
   - Set BOOTPIN_CONFIG.BMSP2 to 0xFF
   - Set BOOTPIN_CONFIG.KEY to 0x5A for boot ROM to treat these register bits as valid and use the custom boot table.
2. Program the BOOTDEF location options for the device. This essentially sets up a device-specific boot mode table. Refer to Section 4.7.7 for valid BOOTDEF values to set in the table.
   - Set BOOTDEF.BOOTDEF0 to 0x02 for CAN booting. This sets CAN boot to boot table index 0.
   - Set BOOTDEF.BOOTDEF1 to 0x03 for booting to Flash (entry address option 0). This sets Flash boot to boot table index 1.

Table 4-10. One Boot Pin Boot Table Result

Boot Mode Table Number	Boot Mode
0	CAN Boot (0x02)
1	Flash Boot (0x03)

4.4.3.3 Three Boot Mode Select Pins

This use-case demonstrates a scenario for an application using three boot mode select pins to select between various boot modes in the custom boot table.

1. Program the BOOTPIN_CONFIG location in OTP as follows:
   - Set BOOTPIN_CONFIG.BMSP0 to a user specified GPIO, such as 0x0 for GPIO0
   - Set BOOTPIN_CONFIG.BMSP1 to a user specified GPIO, such as 0x1 for GPIO1
   - Set BOOTPIN_CONFIG.BMSP2 to a user specified GPIO, such as 0x2 for GPIO2
   - Set BOOTPIN_CONFIG.KEY to 0x5A for boot ROM to treat these register bits as valid and use the custom boot table.
2. Program the BOOTDEF location options for the device. This essentially sets up a device-specific boot mode table. Refer to Section 4.7.7 for valid BOOTDEF values to set in the table.
   - Set BOOTDEF.BOOTDEF0 to 0x02 for CAN booting. This sets CAN boot to boot table index 0.
   - Set BOOTDEF.BOOTDEF1 to 0x03 for booting to Flash. This sets Flash boot to boot table index 1.
   - Set BOOTDEF.BOOTDEF2 to 0x01 for booting to UART booting. This sets UART boot to boot table index 2.
   - Set BOOTDEF.BOOTDEF3 to 0x66 for SPI booting (alternate GPIO option 3). This sets SPI boot to boot table index 3.
   - Set BOOTDEF.BOOTDEF4 to 0x05 for booting to RAM. This sets RAM boot to boot table index 4.

Table 4-11. Three Boot Pins Boot Table Result

Boot Mode Table Number	Boot Mode
0	CAN Boot (0x02)
1	Flash Boot (0x03)
2	UART Boot (0x01)
3	SPI - Alt3 (0x66)
4	RAM Boot (0x05)
5, 6, 7	Not used in this example

4.5 Device Boot Flow Diagrams

This section details the C29 CPU boot flow diagrams for standalone and emulation boot flows.

4.5.1 Device Boot Flow

The following flow diagrams describe how the device boots up after PORESETn. HSM starts up first and then releases reset to CPU1. Detailed CPU1 boot flow is explained in later sections and for detailed HSM boot flow, refer to the *F29x Hardware Security Manager (HSM) User's Guide*.

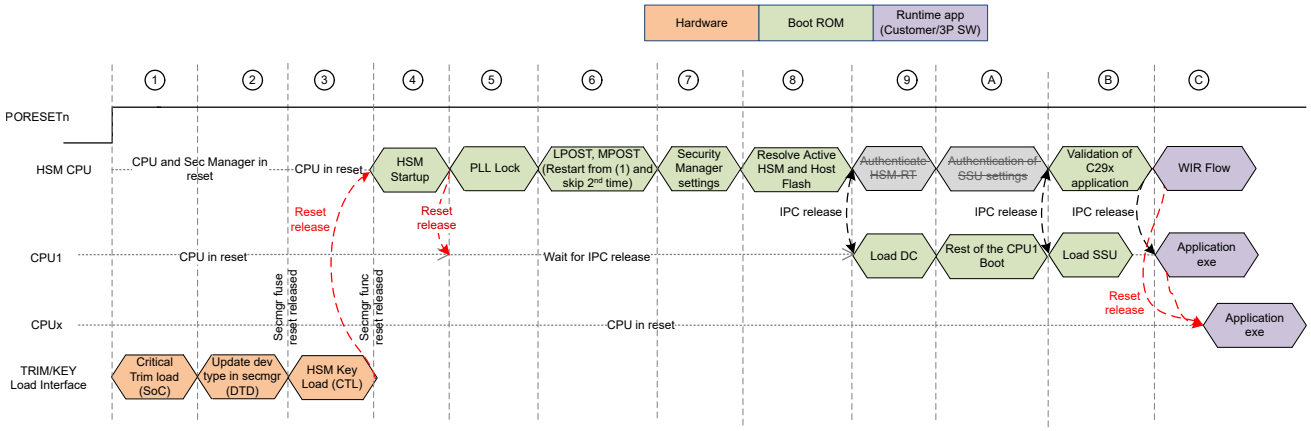


Figure 4-1. HS-FS Device Boot Flow Diagram

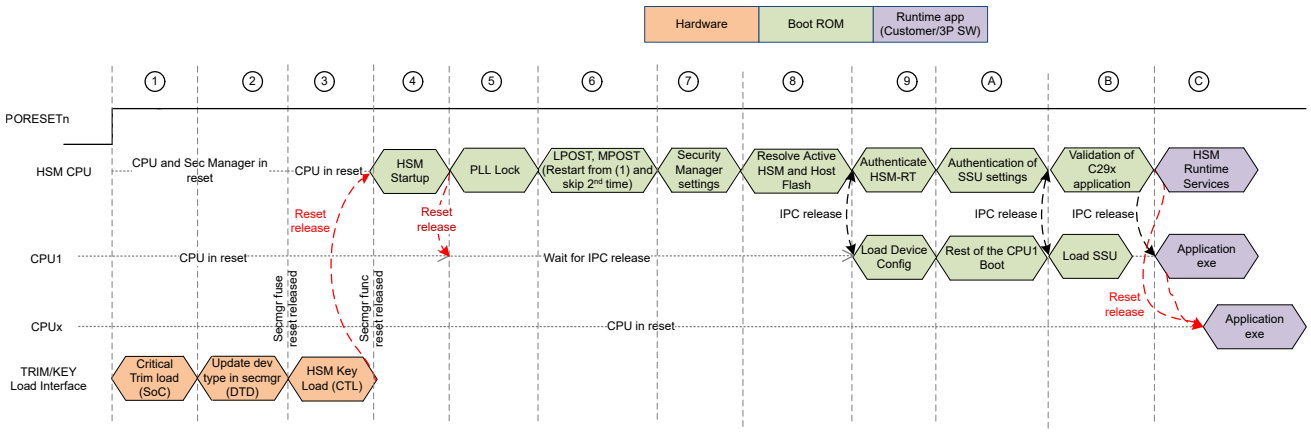


Figure 4-2. HS-SE Secure Boot Flow Diagram

4.5.2 CPU1 Boot Flow

Upon reset, CPU1 follows the boot flow shown in Figure 4-3. Depending on whether a JTAG debugger is connected to the device, the CPU1 either continues booting following the emulation boot flow or the standalone boot flow.

Note

Boot on reset (BOR) follows same flow as power on reset (POR).

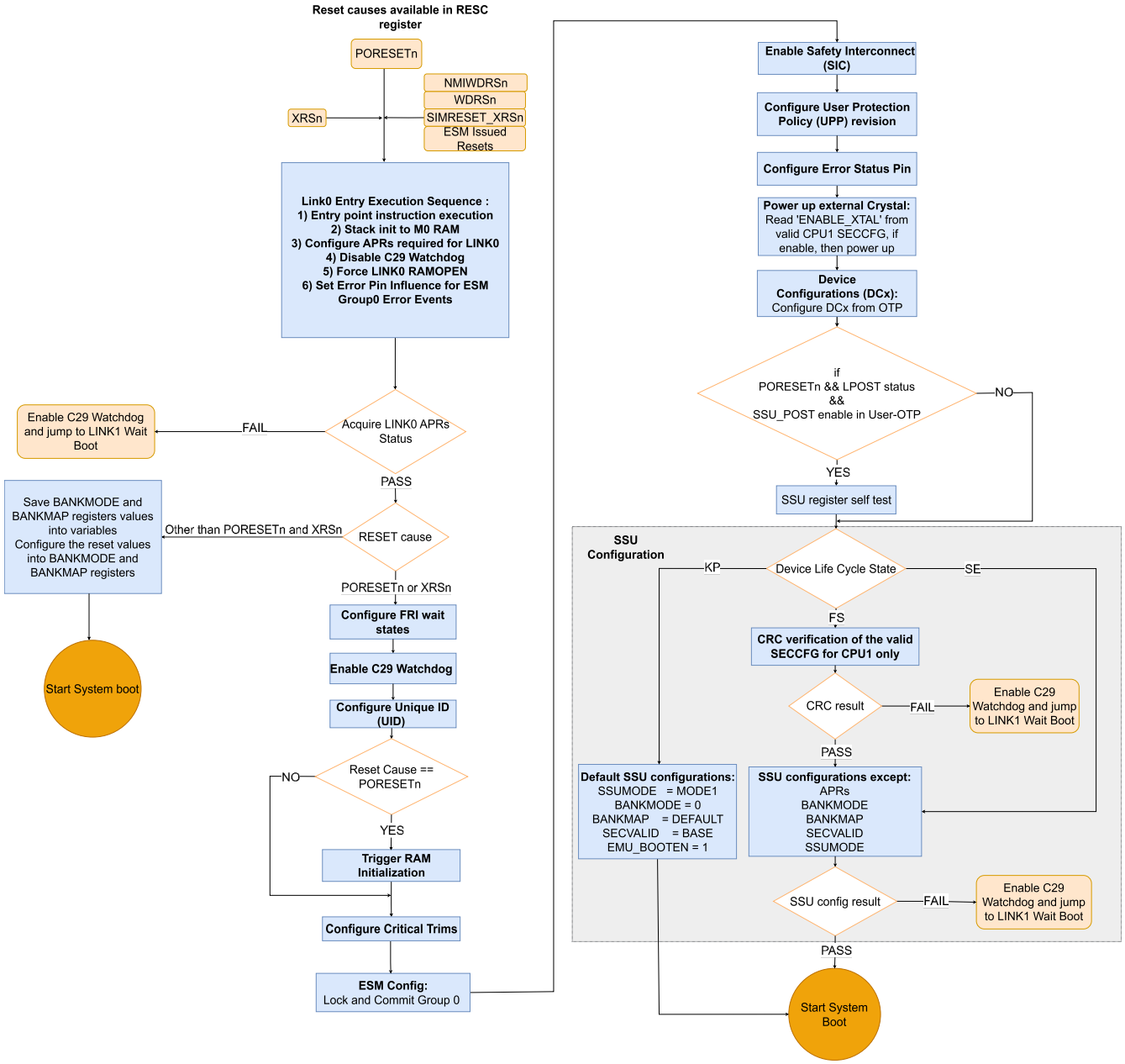


Figure 4-3. Device Boot Flow from Reset to System Boot

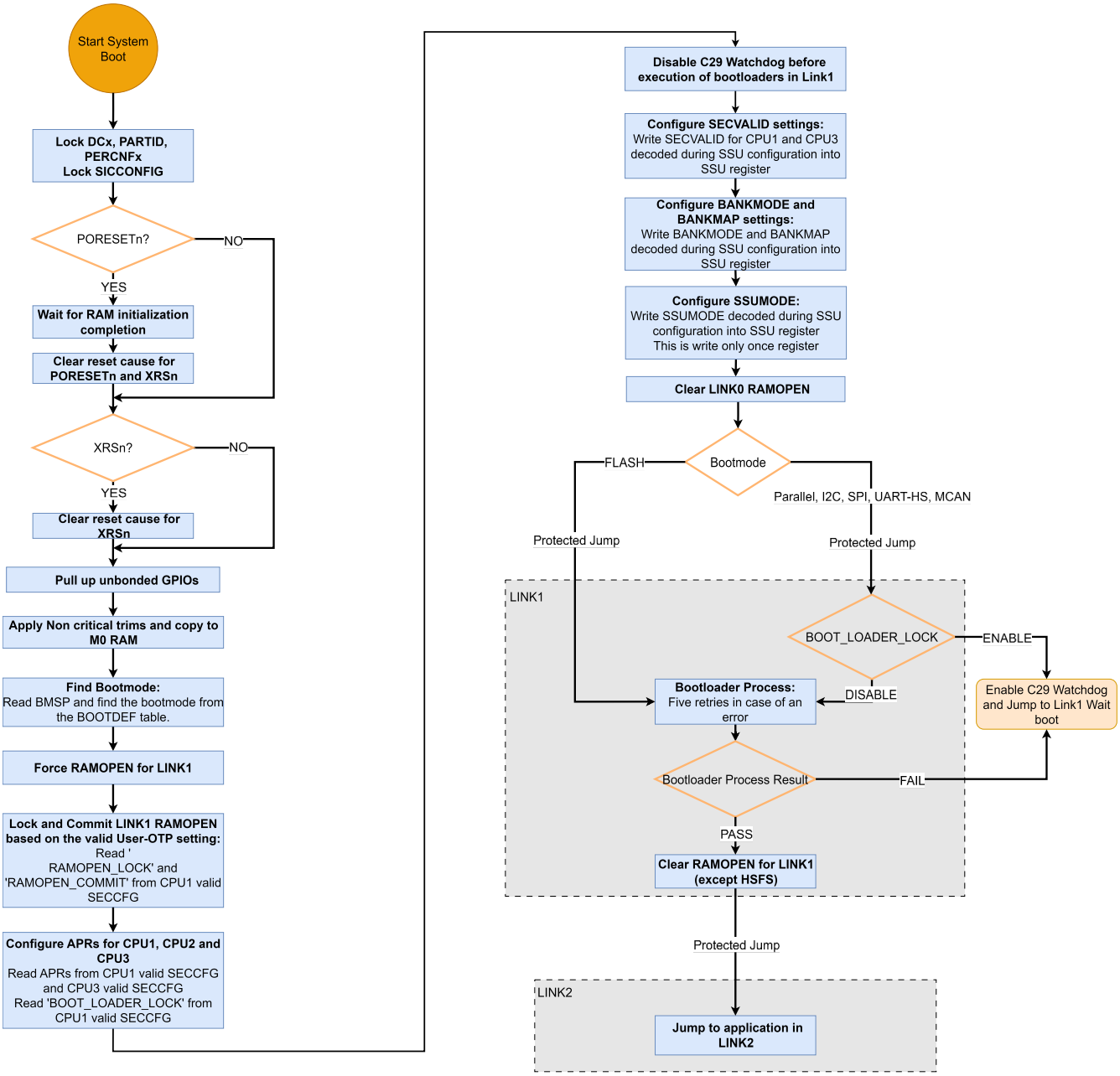


Figure 4-4. System Boot Flow to Application Code

4.5.3 Emulation Boot Flow

Figure 4-5 shows the emulation boot flow when JTAG debugger is connected and emulation boot is enabled in SECCFG User OTP.

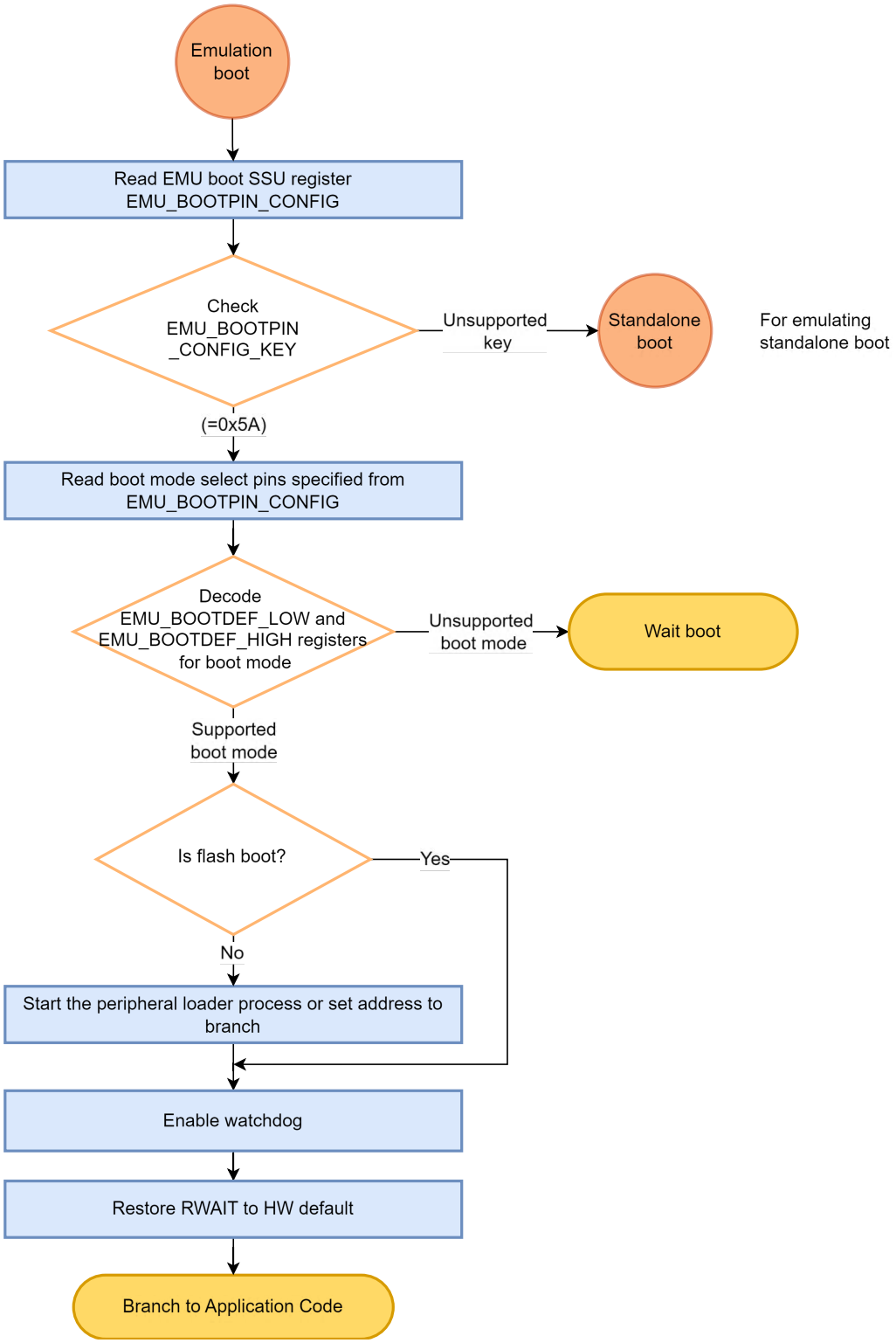


Figure 4-5. Emulation Boot Flow

4.5.4 Standalone Boot Flow

Section 4.5.4 shows the standalone boot flow for CPU1 when no JTAG debugger is connected to the device.

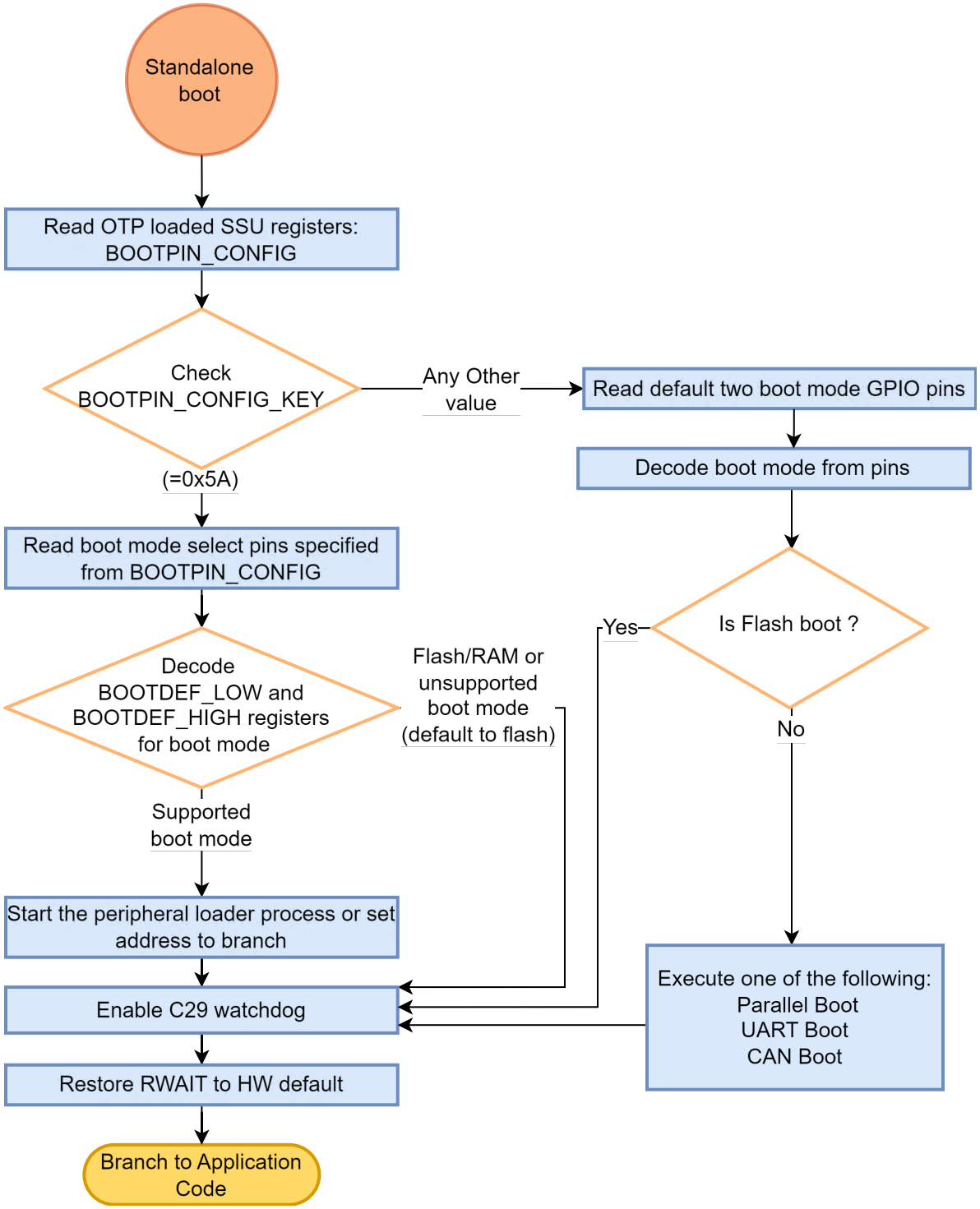


Figure 4-6. Standalone Boot Flow

4.6 Device Reset and Exception Handling

4.6.1 Reset Causes and Handling

Table 4-12 explains the actions each boot ROM performs upon reset for a specific reset cause. The boot ROM flow only explicitly checks for POR and XRS reset cause flags.

Table 4-12. Boot ROM Reset Causes and Actions

Reset Source	CPU1 Boot ROM Action
Power on Reset (POR)	1. Default boot flow 2. RAM initialization
External Reset (XRS) Includes: • CPU1 Watchdog Reset • NMI Watchdog Reset (Issued by ESM) • SIMRESET XRSn • ESM Critical Priority Output (ESMCPUxCRITICAL,ESMRESET) • ECAT RESET	1. Default boot flow
Debugger Reset	1. No change to RAM 2. Save default BANKMAP and BANKMODE

4.6.2 Exceptions and Interrupts Handling

Table 4-13 explains the actions that the boot ROM performs, if any exceptions occur during boot. The philosophy of boot ROM is to try and start the application and log the error.

The boot ROM sets up the default NMI handlers and enables NMI on every reset type. When NMI is triggered on any of the Group0 error event input to CPU1 ESM, the error status pin is driven LOW. The interval for which the error pin remains LOW is determined by the count programmed into SYSTEM ESM instance PIN_CNTR_PRE register. The error status pin GPIO number is read from the SECCFG and configured as error status pin

Table 4-13. Boot ROM Exceptions and Actions

Link	Exception Event Source	CPU1 Boot ROM Action	Event Logged
LINK0 and LINK1	NMI - Uncorrectable ECC Errors	Reset the device	Yes
LINK0 and LINK1	NMI - Other than Uncorrectable ECC Errors	Enable C29 Watchdog and Jump to Link1 and wait in while(1) loop	Yes
LINK0	Critical Trim Loading	Reset the device	Yes
LINK0	LINK0 SSU APR - Failure to acquire APR's for LINK0 execution	Enable C29 Watchdog and Jump to Link1 and wait in while(1) loop	Yes
LINK0	CPU1 valid SECCFG CRC check mismatch	Enable C29 Watchdog and Jump to Link1 and wait in while(1) loop	Yes
LINK0	Error during SSU configuration, SSU self test or SSU APR configurations	Enable C29 Watchdog and Jump to Link1 and wait in while(1) loop	Yes
LINK1	Error during Certificate parsing	Enter Wait Boot	Yes
LINK1	HSM authentication result is "RETRY"	Enter Wait Boot	Yes

4.7 Boot ROM Description

This section explains the details regarding the device boot ROMs.

4.7.1 Boot ROM Configuration Registers

The boot ROM code involves several memory addresses and registers used during execution. There are two sets of configurations; one for emulation and one for standalone boot flow. The emulation locations located in SSU registers - SSU_GEN_REGS, SSU_CPU1_CFG_REGS, SSU_CPU2_CFG_REGS and SSU_CPU3_CFG_REGS emulate the OTP configurations and can be written to as many times as needed. The user configurable SECCFG OTP locations used in the standalone boot flow program the device OTP and hence can only be written once. Table 4-14 details these locations. For bit field configuration details of BOOTPIN_CONFIG and BOOTDEF, see Section 4.4.1 and Section 4.4.2, respectively.

Table 4-14. Boot ROM Registers

Boot Flow	SSU Register Name	OTP SECCFG Name	Register Address	User OTP Address
Emulation	EMU_BOOTPIN_CONFIG		0x3008_0030	-
	EMU_BOOTDEF_LOW		0x3008_1028	-
	EMU_BOOTDEF_HIGH		0x3008_102C	-
Standalone		BOOTPIN_CONFIG	-	0x10D8_17A8
		BOOTDEF_LOW	-	0x10D8_17C8
		BOOTDEF_HIGH	-	0x10D8_17CC

4.7.1.1 MPOST and LPOST Configurations

Using Sysconfig-System Security tab, the users can load the appropriate settings for MPOST and LPOST executions as needed.

4.7.2 Entry Points

This sections gives details about the entry point addresses for various boot modes. These entry points direct the boot ROM what address to branch to at the end of booting as per the selected boot mode.

Table 4-15 give details about the entry point addresses for the Flash boot mode and RAM boot mode.

Table 4-15. Entry Point Addresses for CPU1

Boot Mode	Option	BOOTDEFx Value	Flash Read Port	Address	Packages Supported
Flash	0	0x03	FRI1 RP0	0x1000_1000	All
RAM	0	0x05	N/A	0x2010_0000 (LPA0 SRAM)	All
RAM	1	0x25	N/A	0x2011_0000 (CPA0 SRAM)	All

4.7.3 Wait Points

The wait mode puts the CPU in a loop in the boot ROM code and does not branch to the user application code. The device can enter the wait boot mode either through manually being set or because of some issue during boot up. Using the wait boot mode is recommended when using a debugger to avoid any JTAG issues. There is an infinite for loop when entered in wait boot so user can easily halt CPU and load code when connected using debugger.

Table 4-16. Wait Boot Options

Option	BOOTDEFx Value	Watchdog Status	Package Supported
0 (default)	0x04	Disabled	All

During boot ROM execution, there are situations where the CPU can enter a wait loop in the code. This state can occur for a variety of reasons.

Table 4-17 details the address ranges that the CPU PC register value reaches to if the CPU has entered one of these instances.

Following are the actions for entering wait boot mode:

- Wait boot is set by the user as the boot mode.
- Boot mode is unrecognizable and a debugger is connected to the device.
- An error occurs during emulation boot and the boot mode pins are decoded with a value not recognized as a valid boot mode.

Table 4-17. Wait Point Addresses

Program Counter Address Range	Description
0x1535C	In Wait Boot mode
0x14EB0	In Wait Boot mode

4.7.4 Memory Maps

4.7.4.1 Boot ROM Memory-Maps

Table 4-18 details the ROM memory-map including Link0 and Link1 ROM.

Table 4-18. Boot ROM Memory-Map

Memory	Start Address	Length
Reset Vector	0x0000_0000	0x0040
NMI Vector	0x0000_0040	0x0040
Default Handlers	0x0000_0080	0x0F80
User Data Tables	0x0000_1000	0x7000
Link0 Boot code	0x0000_8000	0x8000
Link1 Boot code	0x0001_0000	0xFFFF

4.7.4.2 Reserved RAM Memory-Maps

Table 4-19 details memory usage in RAM that is reserved for boot ROM to use. These memory sections can be reserved in the user application.

Table 4-19. Reserved RAM Memory-Map

Memory	Description	Origin Address	Length
M0 RAM	Link0 Stack	0x2000_0000	0x800
M0 RAM	Boot ROM Status	0x2000_0800	0x170
M0 RAM	Link0 Data Sections	0x2000_0970	0x5C0
M0 RAM	Non Critical TRIM's	0x2000_0F30	0xCF

4.7.5 ROM Structure and Status Information

This section details the boot ROM structure that can be integrated into an application to use the available ROM code status information and tables.

Table 4-20. ROM Symbol Tables

ROM Structure	Library Name	Location
ROM Code Status Information	driverlib	Under *source/driverlib/bootrom.c and bootrom.h* in MCU_SDK_F29H85x
Twiddle Tables	driverlib	

4.7.6 Boot Modes and Loaders

The available boot modes and bootloaders supported on this device are detailed in this section.

4.7.6.1 Boot Modes

This section details the available boot modes that do not involve a peripheral boot loader. Table 4-21 details the available boot modes that do not involve a peripheral boot loader.

Table 4-21. Boot Mode Availability

Boot Mode	CPU Support
Flash Boot	C29x CPU
RAM Boot	C29x CPU
Wait Boot	C29x CPU

4.7.6.1.1 Flash Boot

Flash boot mode branches to the configured memory address in Flash. Refer to Section 4.7.2 for all the available Flash address options.

4.7.6.1.2 RAM Boot

RAM boot mode branches to the configured memory address in RAM. Refer to Section 4.7.2 for all the available RAM address options.

4.7.6.1.3 Wait Boot

Wait boot mode branches to the memory address as mentioned in Section 4.7.3.

4.7.6.2 Bootloaders

This section details the available boot modes that use a peripheral boot loader.

4.7.6.2.1 SPI Boot Mode

The SPI loader expects an SPI-compatible 16-bit or 24-bit addressable serial EEPROM or serial Flash device to be present on the SPI-A pins as shown in Figure 4-7. The SPI bootloader supports an 8-bit data stream and does not support a 16-bit data stream.

Figure 4-7. Overview of SPI Bootloader Operation

The SPI boot ROM loader initializes the SPI module to interface to a serial SPI EEPROM or Flash. Devices of this type include, but are not limited to, the Xicor X25320 (4Kx8) and Xicor X25256 (32Kx8) SPI serial SPI EEPROMs and the Atmel AT25F1024A serial Flash.

The SPI boot ROM loader initializes the SPI with the following settings: FIFO enabled, 8-bit character length, internal SPICLK controller mode and talk mode, transfer protocol : Mode 1 - rising edge with phase delay, clock phase = 1, polarity = 0 and using 2 Mbps baud rate.

If the download is to be performed from an SPI port on another device, then that device must be set up to operate in the peripheral mode and mimic a serial SPI EEPROM. Immediately after entering the SPI_Boot function, the pin functions for the SPI pins are set to primary and the SPI is initialized.

4.7.6.2.2 I2C Boot Mode

The I2C bootloader expects an 8-bit wide I2C-compatible EEPROM device to be present at address 0x50 on the I2C-A bus as shown in Figure 4-8. The EEPROM must adhere to conventional I2C EEPROM protocol, as described in this section, with a 16-bit base address architecture.

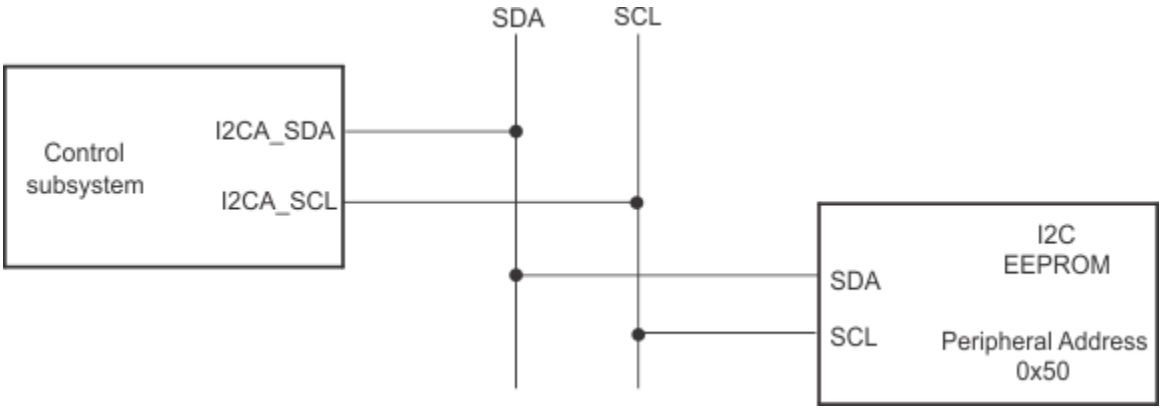


Figure 4-8. EEPROM Device at Address 0x50

If the download is to be performed from a device other than an EEPROM, then that device must be set up to operate in the target mode and mimic the I2C EEPROM. Immediately after entering the I2C boot function, the GPIO pins are configured for I2C-A operation and the I2C is initialized. The following requirements must be met when booting from the I2C module:

- The input frequency to the device must be in the appropriate range.
- The EEPROM must be at target address 0x50.

The bit-period prescalers (I2CCLKH and I2CCLKL) are configured by the bootloader to run the I2C at a 50 percent duty cycle at 100kHz bit rate (standard I2C mode) when the system clock is 10MHz.

Arbitration, bus busy, and target signals are not checked. Therefore, no other controller is allowed to control the bus during this initialization phase. If the application requires another controller during I2C boot mode, that controller must be configured to hold off sending any I2C messages until the application software signals that the software is past the bootloader portion of initialization.

The non-acknowledgment bit is checked only during the first message sent to initialize the EEPROM base address. This is to make sure that an EEPROM is present at address 0x50 before continuing. If an EEPROM is not present, the non-acknowledgment bit is not checked during the address phase of the data read messages. If a non-acknowledgment is received during the data read messages, the I2C bus hangs.

4.7.6.2.3 Parallel Boot Mode

The parallel general-purpose I/O (GPIO) boot mode asynchronously transfers code from the host to C29x internal memory. Each value is 8-bits long and follows the same data flow as outlined in Figure 4-9.

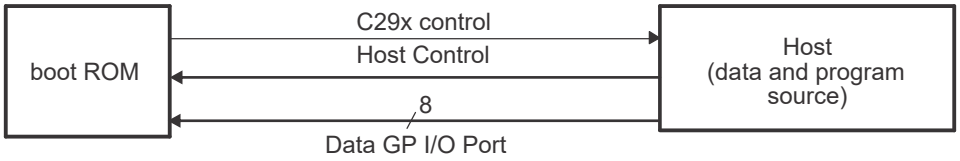


Figure 4-9. Overview of Parallel GPIO Bootloader Operation

The control subsystem communicates with the external host device by polling/driving the Host Control and C29x control lines. The handshake protocol shown in Figure 4-10 must be used to successfully transfer each word using GPIO [D0:D7]. This protocol is very robust and allows for a slower or faster host to communicate with the controller subsystem.

Two consecutive 8-bit words are read to form a single 16-bit word. The least-significant byte (LSB) is read first followed by the most-significant byte (MSB). In this case, data is read from GPIO[D0:D7].

The device first signals the host that the device is ready to begin data transfer by pulling the C29x control pin low. The host load then initiates the data transfer by pulling the DSP control pin low. The complete handshake protocol is shown in Figure 4-10.

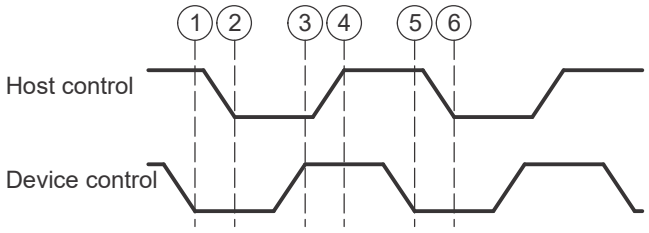


Figure 4-10. Parallel GPIO Bootloader Handshake Protocol

1. The device indicates the device is ready to start receiving data by pulling the C29x control pin low.
2. The bootloader waits until the host puts data on GPIO [D0:D7]. The host signals to the device that data is ready by pulling the host control pin low.
3. The device reads the data and signals the host that the read is complete by pulling the C29x control pin high.
4. The bootloader waits until the host acknowledges the device by pulling the host control pin high.
5. The device again indicates the device is ready for more data by pulling the C29x control pin low.

This process is repeated for each data value to be sent.

Figure 4-11 shows the transfer flow from the host side. The operating speed of the CPU and host are not critical in this mode as the host waits for the device and the device in turn waits for the host. In this manner, the protocol works with both a host running faster and a host running slower than the device.

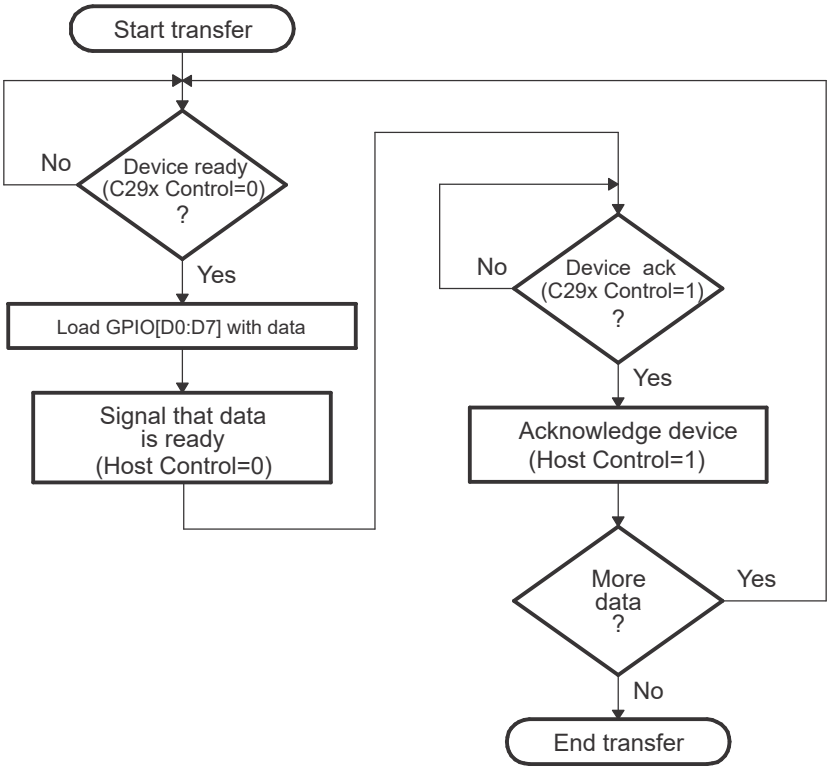


Figure 4-11. Parallel GPIO Mode - Host Transfer Flow

4.7.6.2.4 CAN Boot Mode

The CAN bootloader asynchronously transfers code from CAN-A to internal memory as shown in Figure 4-12. The host can be any CAN node. The communication is first done with 11-bit standard identifiers (with a MSGID of 0x1) using two bytes per data frame.

Note

MCAN and CAN boot can be used with an external oscillator with a 25MHz and 20MHz frequency.

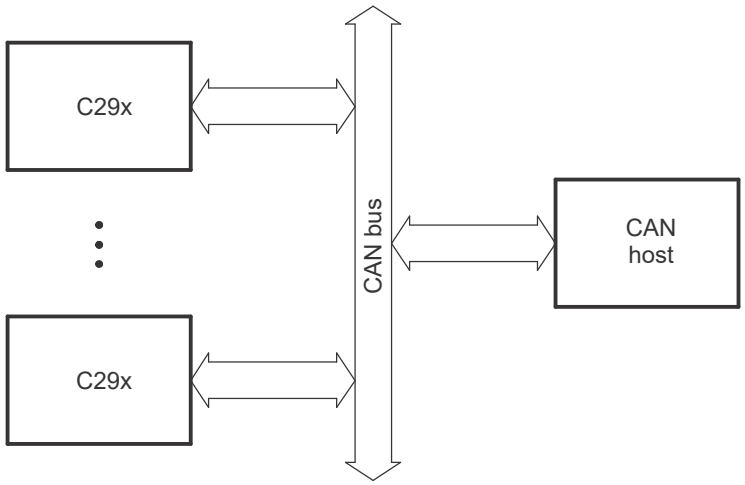


Figure 4-12. Overview of CAN-A Bootloader Operation

The bit timing registers are programmed in such a way that a 100kbps bit rate is achieved with a 20MHz external oscillator, as shown in Table 4-22.

Table 4-22. Bit-Rate Value for External Oscillators

OSCCLK	SYSCLK	Bit Rate
20MHz	200MHz	100kbps
25MHz	200MHz	100kbps

The bootROM configures and locks PLL at 200MHz and also configures CAN bit clock at 20MHz using XTAL oscillator as clock source if detected; otherwise, defaults to INTOSC2.

Note

The CAN boot loader uses XTAL as the bit clock source and INTOSC2 as the system clock source.

Mailbox 1 is programmed with a standard MSGID of 0x1 for bootloader communication. The CAN host transmits only two bytes at a time, LSB first and MSB next. For example, to transmit the word 0x08AA to the device, transmit AA first, followed by 08.

4.7.6.2.5 CAN-FD Boot Mode

The CAN-FD bootloader asynchronously transfers code from CAN-FD to internal memory and follows same bootloader execution flow as Section 4.7.6.2.4. The host can be any CAN-FD node. The communication is first done with 11-bit standard identifiers (with a MSGID of 0x1) using two bytes per data frame. The CAN-FD bootloader uses a fixed 64-byte payload size and default bit rate of 1Mbps for nominal phase and 2Mbps for data phase.

Mailbox 1 is programmed with a standard MSGID of 0x1 for boot-loader communication. The CAN-FD host transmits only two bytes at a time, LSB first and MSB next. For example, to transmit the word 0x08AA to the device, transmit AA first, followed by 08. The program flow of the CAN-FD bootloader is identical to the CAN bootloader.

4.7.6.2.6 UART Boot Mode

The UART bootloader is a peripheral bootloader that receives data and program over the UARTA RX line.

After power-on, when the host is connected and the device is set in UART boot mode, the controller (MCU) sends the SOCID on the UART TX line. Host can start sending the actual image data once the SOCID is received. After each data transfer, the bootloader echo's back the 8-bit character received to the host. This allows the host to check that each character was received by the bootloader.

The SOCID is sent out to the host on every cold/warm reset. The SOCID is also copied to M0 RAM (at address 0x2000 0980), which is transmitted over the UART to the host.

The UART bootloader supports 1 stop bit, no parity and 8-bit character transfers at baud rate of 115200bps.

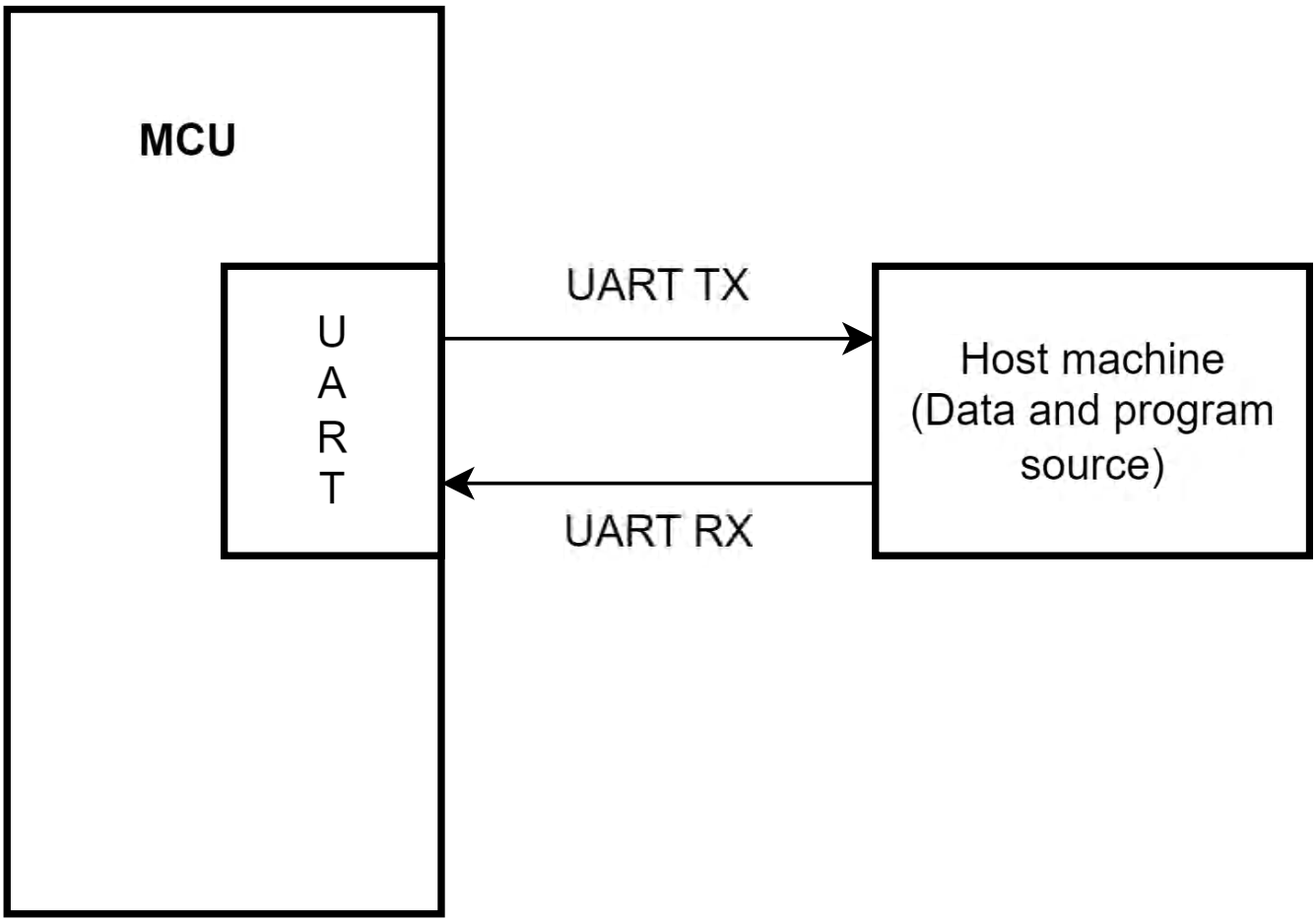


Figure 4-13. UART Boot Mode

4.7.7 GPIO Assignments

This section details the GPIOs and boot option values used for boot mode set in the BOOT_DEF memory location located at BOOTDEF_LOW and BOOTDEF_HIGH. Refer to Section 4.4.2 on how to configure BOOT_DEFx. When selecting a boot mode option, make sure to verify that the necessary pins are available in the pin mux options for the specific device package being used.

Default boot mode GPIO pins:

- Boot mode pin 0 - GPIO84
- Boot mode pin 1 - GPIO72

Guidelines on boot pin selection:

- Avoid pins that have PWM functionality.
- Cannot be analog or USB pins.
- Boot mode select pins and default boot peripheral pins can be available on all packages.
- Avoid JTAG emulation pins and crystal pins.
- Boot mode select pins can be inputs.
- Pins cannot have PHY bootstrap functionality.

Table 4-23. Parallel Boot Options

Option	BOOTDEF Value	D0-D7 GPIO	C29x (DSP) Control GPIO	Host Control GPIO	Package Supported
0 (default)	0x00	D0 - GPIO0 D1 - GPIO1 D2 - GPIO2 D3 - GPIO3 D4 - GPIO4 D5 - GPIO10 D6 - GPIO11 D7 - GPIO12	GPIO15	GPIO16	All
1	0x20	D0 - GPIO17 D1 - GPIO18 D2 - GPIO22 D3 - GPIO23 D4 - GPIO25 D5 - GPIO26 D6 - GPIO29 D7 - GPIO30	GPIO4	GPIO5	All

Table 4-24. UART Boot Options

Option	BOOTDEF Value	TX	RX	Package Supported
0	0x01	GPIO42	GPIO43	All
1	0x21	GPIO38	GPIO39	176-QFP, 256-BGA
2	0x41	GPIO2	GPIO3	All
3	0x61	GPIO38	GPIO3	All
4	0x81	GPIO84	GPIO85	256-BGA

Table 4-25. CAN Boot Options

Option	BOOTDEF Value	CANTXA GPIO	CANRXA GPIO	Package Supported
0 (default)	0x02	GPIO64	GPIO65	All
1	0x22	GPIO234	GPIO235	144-QFP, 176-QFP, 256-BGA
3	0x42	GPIO64	GPIO235	144-QFP, 176-QFP, 256-BGA
4	0x62	GPIO234	GPIO65	144-QFP, 176-QFP, 256-BGA

Table 4-26. SPI Boot Options

Option	BOOTDEF Value	SPIPICOA	SPIPOCIA	SPICLKA	SPISTEA	Package Supported
0	0x06	GPIO58	GPIO59	GPIO60	GPIO61	All
1	0x26	GPIO16	GPIO17	GPIO60	GPIO19	144-QFP, 176-QFP, 256-BGA
2	0x46	GPIO32	GPIO33	GPIO34	GPIO35	256-BGA
3	0x66	GPIO54	GPIO55	GPIO56	GPIO57	176-QFP, 256-BGA

Table 4-27. I2C Boot Options

Option	BOOTDEF Value	SDAA GPIO	SCLA GPIO	Package Supported
0	0x07	GPIO0	GPIO1	All
1	0x27	GPIO32	GPIO33	256-BGA
2	0x47	GPIO42	GPIO43	All
3	0x67	GPIO56	GPIO57	144-QFP, 176-QFP, 256-BGA

Table 4-28. CAN-FD Boot Options

Option	BOOTDEF Value	MCAN TX	MCAN RX	Package Supported
0	0x08	GPIO64	GPIO65	All
1	0x28	GPIO234	GPIO235	144-QFP, 176-QFP, 256-BGA
2	0x48	GPIO64	GPIO235	144-QFP, 176-QFP, 256-BGA
3	0x68	GPIO234	GPIO65	144-QFP, 176-QFP, 256-BGA

4.7.8 HSM and C29 ROM Task Ownership and Interactions

HSM (M4) has the primary ownership of the device at boot up and does the following tasks:

1. PLL Initialization and Lock
2. Run MPOST based on input from SECCFG settings input by user
3. Run LPOST based on input from SECCFG settings input by user
4. IPC Initialization and C29 CPU1 reset release
5. C29 Application Image authentication
6. C29 SBL validation and integrity check
7. HSMRT validation and integrity check

C29 SBL (Secondary Bootloader) authentication process:

1. Initialize Boot Peripheral to receive the image
2. Copies image from host and loads into LDA RAM
3. After copying, C29 sends IPC message to HSM
4. HSM starts certificate validation and image integrity checks
5. HSM sends ACK back to C29 CPU1 and copies image from LDA to LPA RAM in case image authentication succeeds
6. If the image authentication fails, then HSM retries the process again for a maximum of 5 times

HSMRT (HSM Realtime App) authentication process:

1. SBL downloads HSMRT from the peripheral bootloader chosen by user
2. SBL copies HSMRT image at LDA RAM
3. C29CPU sends IPC message to HSM after copying image
4. HSM protects LDA memory and starts certificate and image validation
5. HSM branches to LDA and releases protection in case of successful validation
6. In case of failure, send NACK message to C29 CPU1

4.7.8.1 Application Authentication by HSM

The application image either present on Flash or downloaded to RAM by peripheral bootloader is authenticated by HSM.

In case of peripheral bootloader, the following procedure is followed:

1. The certificate is downloaded to LDA7 RAM (0x200E_0000)
2. The image is downloaded at an offset of maximum size of certificate (4KB) at 0x200E_1000
3. HSM reads the certificate and image and authenticates the image
4. If the image authentication succeeds, then the image is copied to LPA0 RAM (0x20100000)
5. Then LINK1 makes protected jump to LPA0 address and starts executing the image

The HSM and C29 CPU image authentication process is also summarized in Figure 4-14.

Table 4-29. Authentication Process Error Handling

Error	Handling
Authentication Failure	Retry for 5 times and if error still persists then enter wait boot

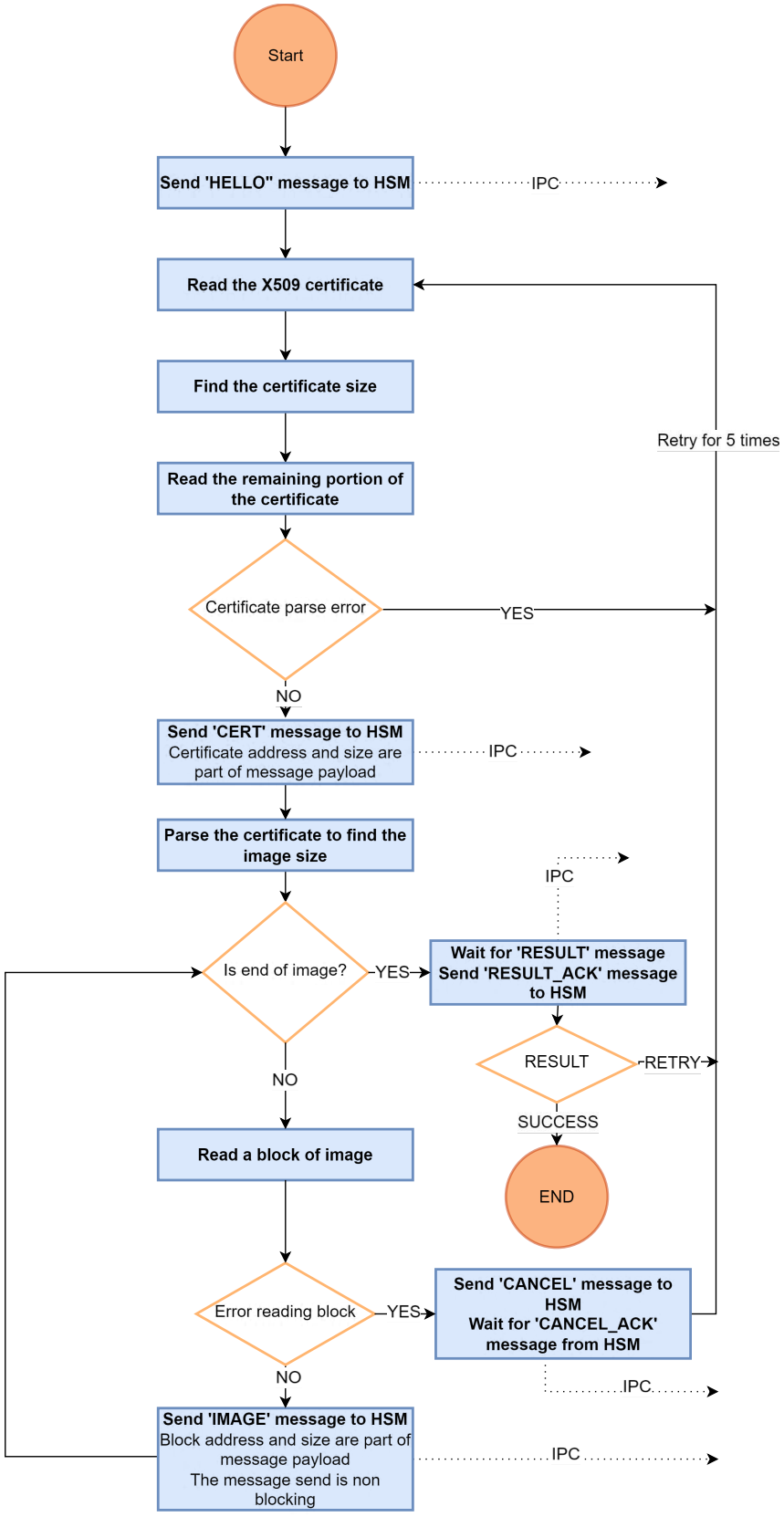


Figure 4-14. Image Authentication by HSM

4.7.9 Boot Status Information

Boot ROM keeps a record of the various actions and events that occur during boot ROM execution. The reason for this is because NMI and other exceptions are enabled by default in the device and must be handled accordingly. Boot ROM stores the boot status information in a RAM location so that the user application can read this boot status and take the necessary actions per application's needs to handle these events.

4.7.9.1 Booting Status

Boot ROM health and booting status for CPU1 is written to a 32-bit address in the respective M0 RAM. This status is cleared on a POR or XRS reset. The previous status is retained on any other reset. For example, clear the status before performing a debugger device reset to view the latest boot ROM actions reflected in the status.

Table 4-30. Boot Information Parameters Address

Parameter	Address	Description
Link 0 Start Cycle Count	0x2000_0800	64 bit Link0 Start Cycle Count
Link 0 End Cycle Count	0x2000_0808	64 bit Link0 End Cycle Count
Reset Count	0x2000_0810	Reset Count
Reset Cause	0x2000_0814	Reset Cause Register(RESC) saved before cleared in the boot code
Sysclk frequency	0x2000_0818	Sysclk Frequency in Hz
ECC Error Addr DR1	0x2000_081C	ECC Error Address on DR1 Port
ECC Error Addr DR2	0x2000_0820	ECC Error Address on DR2 Port
Boot Mode	0x2000_0824	Boot Mode decoded by boot code is saved here
Boot Loader Lock	0x2000_0828	Lock is enabled if value = 0xC9 Disabled if value = 0xFF
Device Life Cycle	0x2000_082C	Device Life cycle Value
Link0 Error Id	0x2000_0830	Error ID for Error occurring in Link0 Boot Code

Table 4-31. Link0 Error ID Value Description

Link0 Error ID	Value
No Error	0xF487_DA78
APR Configuration Error	0x2AC6_9B4F
SSU POST Error	0xADF5_3EF6
SSU Configuration Error	0x57DE_827C
NMI Error	0xCDF5_847D
SECCFG APR Error	0x69D4_73BE
Watchdog SYNCBUSY Error	0xFE3B_C723
XTAL SYNCBUSY Error	0x9BC4_EA4C

Table 4-32. Device Life Cycle Value Description

Device Life Cycle	Value
HS-KP	0x2D3B_68E5
HS-FS	0xA540_1C71
HS-SE	0xA4B4_5974

Table 4-33. Boot Flow Execution Status

Description	Start Address
Boot Flow Execution Status Address	0x2000_0834

Table 4-34 describes the boot flow execution status bit field descriptions.

Table 4-34. Boot Flow Execution Status Descriptions

Bit Field Name	Description
RAM Initialization Status	Each entry is a 2-bit field status where the value description is: 01 - Did not Run 10 - PASS 11 - FAIL
Reserved	
Error Status Pin Configuration Status	
Reserved	
UID Configuration Status	
FRI Wait State Configuration Status	
WatchDog Enable Status	
Reserved	
Reserved	
Reserved	
Reserved	
Reserved	
Device Configuration Status	
Lock DCx Status	
Reset Cause Clear Execution Status	
Reserved	
ESM Lock and Commit Status	
UPP Revision Configuration Status	
XTAL Enable Execution Status	
SSU Self Test Status	
SSU Initialization	
Reserved	
Reserved	
Reserved	
Reserved	
Reserved	

Table 4-35. CPU1 Boot Error Status Address

Description	Address
ESM RAW Status	0x2000_0868
CPU1 PR Error Aggregator High Priority Error address	0x2000_086C
CPU1 PR Error Aggregator Low Priority Error address	0x2000_0870
CPU1 PR Error Aggregator Error Type	0x2000_0874
CPU1 PR Error Aggregator PC value	0x2000_0878
CPU1 DR1 Error Aggregator High Priority Error address	0x2000_087C
CPU1 DR1 Error Aggregator Low Priority Error address	0x2000_0880
CPU1 DR1 Error Aggregator Error Type	0x2000_0884
CPU1 DR1 Error Aggregator PC value	0x2000_0888
CPU1 DR2 Error Aggregator High Priority Error address	0x2000_088C
CPU1 DR2 Error Aggregator Low Priority Error address	0x2000_0890
CPU1 DR2 Error Aggregator Error Type	0x2000_0894
CPU1 DR2 Error Aggregator PC value	0x2000_0898
CPU1 DW Error Aggregator High Priority Error address	0x2000_089C
CPU1 DW Error Aggregator Low Priority Error address	0x2000_08A0
CPU1 DW Error Aggregator Error Type	0x2000_08A4
CPU1 DW Error Aggregator PC value	0x2000_08A8
CPU1 INT Error Aggregator High Priority Error address	0x2000_08AC
CPU1 INT Error Aggregator Low Priority Error address	0x2000_08B0
CPU1 INT Error Aggregator Error Type	0x2000_08B4
CPU1 INT Error Aggregator PC value	0x2000_08B8

Table 4-36. Boot SSU Execution Status Address

Description	Address
SSU Execution Status	0x2000_083C
Address of Cpu1 Winning BankMgmt sector	0x2000_0840
Address of Cpu3 Winning BankMgmt sector	0x2000_0844
Address of CPU1 valid SECCFG sector	0x2000_0848
Address of CPU3 valid SECCFG sector	0x2000_084C
Bank Mode	0x2000_0850
CPU1 swap setting	0x2000_0854
CPU3 swap setting	0x2000_0858
CPU1 SECVALID setting	0x2000_085C
CPU3 SECVALID setting	0x2000_0860
SSUMODE setting	0x2000_0864

4.7.10 BootROM Timing

Table 4-37 describes the estimated boot time impact from reset to application code time jump.

Table 4-37. BootROM Timing

Options/Boot Time (time to jump app)	Estimated Time (msec)
HSFS (no MPOST/LPOST)	32
HSFS (with MPOST/LPOST)	64
HSSE (no MPOST/LPOST)	84
HSSE (with MPOST/LPOST)	116

Note

All timing numbers include **20ms** as AutoSAR stack initialization time.

4.8 Software

4.8.1 BOOT Examples

NOTE: These examples are located in the C2000Ware installation at the following location:
C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/boot

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

Chapter 5

Lockstep Compare Module (LCM)

This chapter describes the Lockstep Compare Module (LCM).

5.1 Introduction

Hardware module integrity during run-time is a critical functional safety requirement. Hardware Redundancy implemented by the lockstep CPU architecture (two CPUs executing the same function and the output of the CPUs are continuously compared) is a proven method for achieving high diagnostic coverage for both permanent and transient faults. The Lockstep Comparator Module (LCM) is implemented to compare output from the CPU to detect permanent and transient faults.

5.1.1 Features

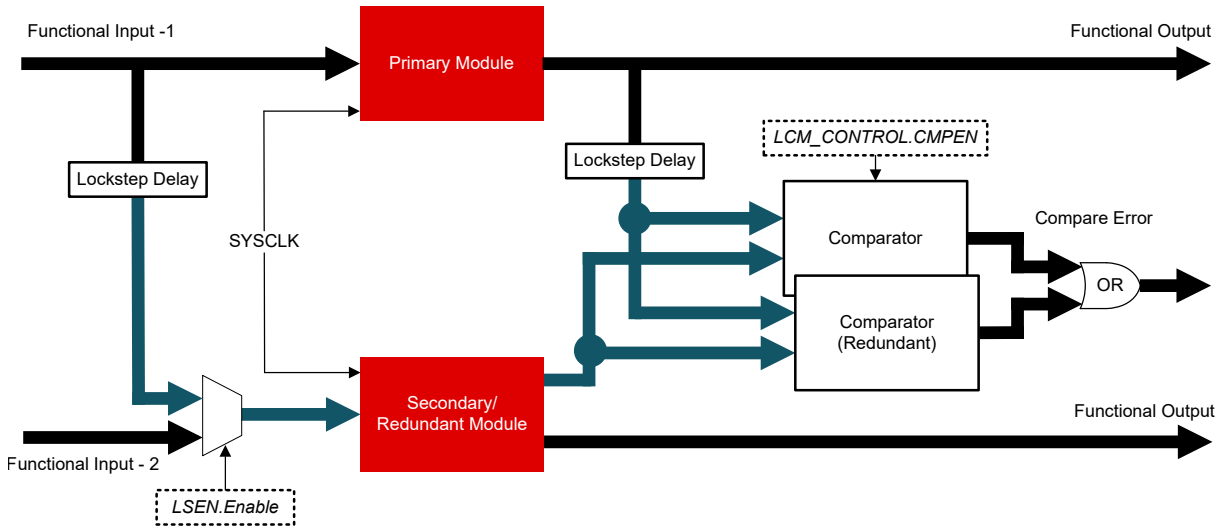
The LCM implements the following features:

- Pipelined architecture
- Redundant comparison
- Self-test capability
  - Match and mismatch test
  - Error forcing capability
- Temporal redundancy: The operation of the two modules is skewed by two cycles to address the issue of common cause failures like failure of clock, power, and so on. This makes sure of temporal redundancy.
- Spatial redundancy: In the lockstep architecture, module instances are redundantly instantiated and the outputs are compared. Redundant instantiation provides spatial redundancy.
- Non-delayed functional output path to provide non-delayed CPU execution for the system (while still having temporal redundancy).
- Register protection of critical memory mapped registers of the module, using a parity scheme.

5.1.2 Block Diagram

Figure 5-1 shows the LCM block diagram.

Figure 5-1. LCM Block Diagram



Note

The *Module* described in this block diagram can be either a CPU (for example, CPU1) or a peripheral (for example, RTDMA) depending on availability for the device.

5.1.3 Lockstep Compare Modules

Following modules are supported with Lockstep comparison feature in this device :

1. CPU1 in lockstep with CPU2
2. CPU1 PIPE in lockstep with CPU2 PIPE
3. RTDMA1 in lockstep with RTDMA2
4. CPU1 Timers (0, 1, and 2) in lockstep with CPU2 Timers (0, 1, and 2)
5. CPU1 ERAD and DLT in lockstep with CPU2 ERAD and DLT, respectively

DEV_CFG_REGS[LSEN] register controls lockstep enable/disable for all the modules that are mentioned above, which means either all of them operate as lockstep or split-lock (operating as independent modules).

Table 5-1. Lockstep versus Split-Lock Configurations

Configuration Type	DEV_CFG_REGS[LSEN] Default : 1	LCM_CONTROL_REGS[CMPEN] Default : 0
Lock-Step	1	1
Split-Lock (Modules operating independently)	0	Not Applicable

5.2 Enabling LCM Comparators

Note

The redundant module is automatically enabled at boot.

To enable the LCM comparators upon device startup or after reset, perform the following steps at the beginning of the user application code:

1. Lock the LSEN configuration using the following bit:
   - DEV_CFG_REGS[DEVCFGLOCK2.LSEN = 1]
2. Enable the comparator for the desired modules using the following bit:
   - LCM_REGS[LCM_CONTROL.CMPEN = 1]
   - Lockstep compare begins the immediate next cycle.

   Note: The lockstep comparison is enabled by this write and cannot be disabled again without a reset.

5.3 LCM Redundant Module Configuration

In systems that do not require the redundant module output for comparison in functional safety requirements, the LCM redundant module can be used for functional output. To change the LCM comparators configuration upon device startup or after reset, the following steps must be taken at the beginning of the user application code:

1. Disable the redundant module using the following bit:
   - DEV_CFG_REGS[LSEN.Enable = 0]

   Note: The redundant lockstep module is supplied with a functional input and the comparison must also be disabled at this time. The configuration cannot be altered without a reset.
2. (Optional) Lock the LSEN configuration using the following bit:
   - DEV_CFG_REGS[DEVCFGLOCK2.LSEN = 1]

5.4 LCM Error Handling

Upon an error generated by the LCM module, an error event input is sent to the error signaling module (ESM). The ESM can be configured to take appropriate action that includes error status pin indication, generation of an interrupt or NMI to the CPU. The LCM error can be triggered by any of the following:

- Functional failure: LCM detecting an error (functional failure between the two modules).
- Self-test failure: LCM self-test failure detected (functional failure on one or both of the modules).
- Force Compare Error Request: Running LCM_CONTROL.CMPx_ERR_FORCE to force a compare error.

To determine the specific cause of the LCM error-induced NMI, the LCM_STATUS register bits can be read. Particularly:

- CMP_FAIL
- STPASS
- CMPx_ERR_FORCE_PASS
- CMPx_ERR_FORCE_DONE

For details on system-level NMIs vector locations and interrupt-handling, see Chapter 6.

5.5 Debug Mode with LCM

Lockstep comparison is disabled automatically during debug/emulation mode of the device. However, some items are still available when a debugger is connected:

- Code can continue to be debugged even though lockstep compare is disabled with a debugger connection.
- Self-test logic (match test and mismatch test) is still accessible with debugger connected.
- Clock and inputs to the secondary module are not impacted, and continue to get delayed clock and delayed inputs.

The status of the debugger connection is readable from the status register, specifically the LCM_REGS[LCM_STATUS.DBGCON] bit.

To re-initialize the lockstep compare module (after debugger is disconnected), a reset is required. This reset is called using a system reset requested from the debugger which in turn generates an XRSn in the device.

5.6 Register Parity Error Protection

The following critical LCM registers are protected by a parity scheme:

- LCM_CONTROL
- LCM_STATUS
- LCM_LOCK
- LCM_COMMIT

The parity scheme provides one parity bit per byte of data in the corresponding registers. Updates to any of the constantly-monitored registers causes an update to the parity bit. A single bit fault can therefore immediately flag an error. If the parity check determines a parity error has occurred, a dedicated error output line from the LCM module flags an error to the system.

All register parity errors (from the LCM) are sent as error inputs to the ESM module, refer to Chapter 7 for more details.

Upon a parity error detection, SYSRSN must be asserted and the LCM_REGS[LCM_STATUS_CLEAR] register must then be cleared using the a write of 1 to all appropriate bits in the register.

Details on the self-test capability of the register parity error test are explained in Section 5.7.3.

5.7 Functional Logic

This section describes the various logical blocks within the LCM that contribute to the goal of improved diagnostic coverage.

5.7.1 Comparator Logic

To implement lockstep scheme, a comparator block is needed. The comparator block compares the delayed version of relevant outputs of the primary module and the equivalent outputs of the secondary (redundant) module. This provides immunity towards the common cause failures like loss of power, clock failures, etc.

The LCM has two instantiations of the comparator block to provide redundancy of the comparator block. This enables availability of one comparator block during self-test of the other comparator block and also provides additional failure protection capability for the comparator logic.

Although the lockstep secondary (redundant) module is enabled upon startup, lockstep comparison must be enabled in software.

Once lockstep compare is enabled in software, the comparison is performed continuously every cycle, from the immediate next cycle. Any difference between the primary and secondary modules generates an error signal from the LCM to the SoC. The corresponding register bit is also set (LCM_STATUS.CMP_FAIL).

CPU reset (or any higher-level resets) disables the lockstep comparators, requiring re-enabling of the comparator in software.

5.7.2 Self-Test Logic

The self-test of the comparator has two different modes:

- Match Test
- Mismatch Test

These two tests are run together. Self-test is initiated by setting the appropriate register bit (LCM_CONTROL.STEN). When the self-test is initiated, the two different modes are executed on the two comparators one after the other. A self-test error also triggers error to ESM which can be configured to generate an NMI.

Redundant instantiation of the comparator block allows for one instantiation to be conducting a self-test while the other instantiation is active and performing the comparison check.

Self-test can also be performed before the comparator block is enabled in software.

Execution of the self-test is stopped immediately on failure. If either comparator fails the self-test, a status bit (LCM_STATUS.STPASS) is 0 instead of being set to 1.

During self-test, the LCM_STATUS.STACTIVE bit has a value of 1. Another self-test or compare error force must not be started until completion of the self-test, as indicated by LCM_STATUS.STDONE = 1.

The following subsections describe functionality of each of the individual test modes, Mismatch Test and Match Test.

5.7.2.1 Match Test Mode

The match test mode checks to make sure identical inputs on each comparator provide passing outputs. This makes sure that the comparator inputs and comparators themselves are working correctly and that a fault is successfully propagated to the module output.

This test is executed using two different patterns fed to the two inputs of each comparator (bit) in the comparator block: {0,0} and {1,1}. Both inputs can provide a passing output to the comparator since both patterns are providing identical inputs to both inputs of the comparator. This tests output-stuck-at-one, input-stuck-at-one, and input-stuck-at-zero issues. Table 5-2 shows the test execution sequence for a single comparator in the comparator block.

Table 5-2. Match Test Simplified Example

Clock Cycle	A Input (Primary Module) of Comparator	B Input (Secondary Module) of Comparator	Output
0	0	0	0
1	1	1	0

5.7.2.2 Mismatch Test Mode

The mismatch test mode creates an error output of 1 on each individual comparator (bit) within the comparator block, one at a time. This makes sure that all comparators in the block are working correctly and that a fault is successfully propagated to the module output.

This test is executed using a walking 1s pattern to test for output-stuck-at-zero issues. The walking 1s pattern is where all comparators in the comparator block are zero except for one of the comparators.

For example, the primary module has a 1 at the spot that the secondary module has a 0. This can force an intentional error.

This is repeated for every comparator in the block, with the 1 being set on both modules. The passing scenario for this is that all comparators see a mismatch, flagging a mismatch each iteration. See Figure 5-2 for a simplified illustration of how this is implemented for a comparator block with 8 comparators (not the number used in the actual design).

	Secondary Module Signal								Primary Module Signal								
Cycle #	7 (n)	6	5	4	3	2	1	0	7 (n)	6	5	4	3	2	1	0	Output
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
4	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
6	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
11	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
12	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
13	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
14	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
15 (2n)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 5-2. Mismatch Test Simplified Example

5.7.3 Error Injection Tests

There are two error injection tests available for the LCM:

- Comparator Error Force Test
- Register Parity Error Injection Test

Each error injection test is run individually, as described in the following subsections.

5.7.3.1 Comparator Error Force Test

The LCM has the capability to force a fault to check the error signaling path between the LCM and ESM at the device level (for example, generate NMI, indicate error on ERRORSTS pin or trigger reset.) This test is executed separately from the self-test, and is executed on the primary and redundant comparators individually through separate calls.

Note

- The lockstep comparator must be enabled to execute this test.
- Once this self-test is triggered, another self-test or compare error force request must not be triggered again until the current test is completed, as indicated by LCM_STATUS.STDONE == 1.

Execution of this self-test is started by setting the appropriate bit: LCM_CONTROL.CMPx_ERR_FORCE, where x is the number designator of the redundant comparator to be tested.

Upon execution of this one-cycle-long test, the LCM asserts a lockstep comparison error signal to the device. The normal functional compare fail flag (LCM_STATUS.CMP_FAIL) is not set by this test mode.

The following bits are set by the test:

- LCM_STATUS register:
  - CMPx_ERR_FORCE_DONE bit - 1 when the test is completed.
    - This bit must be cleared before running the test again for comparator "x".
  - CMPx_ERR_FORCE_PASS bit - 1 when the test passes.
    - This bit must be cleared before running the test again for comparator "x".

Because the test can trigger an NMI if configured in ESM on a test pass, the "DONE" flag allows the cause to be determined by either a functional fail or an error forcing test fail.

5.7.3.2 Register Parity Error Test

An error can be injected into the register parity error protection that exists for critical LCM registers, to test for latent faults. This error is inserted by forcing a particular byte to output a failing parity state to the system.

The test can be executed by running setting the LCM_REGS[PARITY_TEST.TESTEN] bits to the appropriate value.

Once the TESTEN register bits are set, the actual registers are no longer accessible in the memory map. Instead, the one bit parity error status values are accessible for every byte in the registers. Parity is computed for every byte, and the corresponding parity pass/fail value is available at the bit 0 location of every byte, in-place. A 0 in the bit 0 location corresponds to a pass; a 1 in the bit 0 location corresponds to a fail.

To actually inject an error, the value 1 can then be written to the bit 0 location of any byte. This inverts the stored parity value, and therefore inject an error into the parity test mechanism. Note that this propagates like an actual parity error in ESM and handled accordingly.

5.8 Software

5.8.1 LCM Registers to Driverlib Functions

Table 5-3. LCM Registers to Driverlib Functions

File	Driverlib Function
REVISION	
-	
CONTROL	
-	
STATUS	
-	
STATUS_CLEAR	
-	
PARITY_TEST	
-	
LOCK	
-	
COMMIT	
-	

5.9 LCM Registers

This section describes the LCM Registers.

5.9.1 LCM Base Address Table

Table 5-4. LCM Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
LCM_REGS [1]	LCM_CPU_BASE	0x3003_2000	-	YES	-	-	-	-	-	YES
LCM_REGS [1]	LCM_DMA_BASE	0x301F_4000	-	YES	YES	YES	-	-	YES	YES

(1) Registers writeable by CPU1.LINK0, CPU1.LINK1, CPU1.LINK2 only. All CPUs can read all registers in all LINKs. Debug write access only allowed if Zone0 or Zone1 are enabled for full debug by all CPUs. Debug reads always allowed. Register Read/Write access by HSM.

5.9.2 LCM_REGS Registers

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

Table 5-5. LCM_REGS Registers

Offset	Acronym	Register Name	Protection
0h	REVISION	IP Revision tie-off value	
8h	LCM_CONTROL	LCM Control configuration	PARITY
20h	LCM_STATUS	LCM status register	PARITY
28h	LCM_STATUS_CLEAR	LCM Status clear register	
68h	PARITY_TEST	Enabling the parity test feature	
70h	LCM_LOCK	LCM lock configuration	PARITY
78h	LCM_COMMIT	LCM commit configuration	PARITY

Complex bit access types are encoded to fit into small table cells. Table 5-6 shows the codes that are used for access types in this section.

Table 5-6. LCM_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

5.9.2.1 REVISION Register (Offset = 0h) [Reset = 40000000h]

REVISION is shown in Figure 5-3 and described in Table 5-7.

Return to the Summary Table.

IP Revision tie-off value

Figure 5-3. REVISION Register

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

Table 5-7. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	This identifies the scheme revision ID register type implemented for this module Reset type: SYSRSn
29-28	RESERVED	R-0	0h	Reserved
27-16	FUNC	R	0h	Functional Release Number Reflects software-compatability. If there is no software compatability, a unique func number is assigned for compatible modules, the same number is maintained. Reset type: SYSRSn
15-11	RESERVED	R	0h	Reserved
10-8	MAJOR	R	0h	Major Revision Number Represents major changes to the module (e.g. entirely new features are added/changed). The major revision number for this module. Reset type: SYSRSn
7-6	CUSTOM	R	0h	Custom Module Number Indicates a special version of the module. May not be supported by standard software. Reset type: SYSRSn
5-0	MINOR	R	0h	Minor Revision Number Represents minor changes to the module (e.g. enhancements to existing features). The minor revision number for this module. Reset type: SYSRSn

5.9.2.2 LCM_CONTROL Register (Offset = 8h) [Reset = 00000000h]

LCM_CONTROL is shown in Figure 5-4 and described in Table 5-8.

Return to the Summary Table.

LCM Control configuration

Figure 5-4. LCM_CONTROL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED		CMP2_ERR_FORCE	RESERVED	CMP1_ERR_FORCE	RESERVED		STEN
R-0h		R-0/W-0h	R-0h	R-0/W-0h	R-0h		R-0/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0/W-0h							
7	6	5	4	3	2	1	0
RESERVED							CMPEN
R-0/W-0h							R/W-0h

Table 5-8. LCM_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	Reserved
21	CMP2_ERR_FORCE	R-0/W	0h	0: configuration is ignored 1: comparator-2 lockstep compare error is forced (i) Once the bit is configured, comprator-2 compare error output will be asserted for one cycle. This feature is used to check the error propagation path from comparator2 compare error output to the observation point defined in system control (ii) The test shall be triggered only after enabling the lockstep feature. (iii) It is not possible to execute this test with debugger connected. (iv) The test cannot be executed if there is pending functional failure or test failure (i.e. test cannot be executed when LCM_STATUS.cmp_fail = 1 or (LCM_STATUS.stpass = 0 and LCM_STATUS.stdone = 1) or (LCM_STATUS.cmp1_err_force_pass = 0 and LCM_STATUS.cmp1_err_force_done = 1) or (LCM_STATUS.cmp2_err_force_pass = 0 and LCM_STATUS.cmp2_err_force_done = 1) (v) LCM_STATUS.cmp2_err_force_done and LCM_STATUS.cmp2_err_force_pass flags need to be cleared before initiating the test a 2nd time. Reset type: SYSRSn
20	RESERVED	R	0h	Reserved

Table 5-8. LCM_CONTROL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	CMP1_ERR_FORCE	R-0/W	0h	0: configuration is ignored 1: comparator-1 lockstep compare error is forced (i) Once the bit is configured, comprator-1 compare error output will be asserted for one cycle. This feature is used to check the error propagation path from comparator1 compare error output to the observation point defined in system control. (ii) The test shall be triggered only after enabling the lockstep feature. (iii) It is not possible to execute this test with debugger connected. (iv) The test cannot be executed if there is pending functional failure or test failure (i.e. test cannot be executed when LCM_STATUS.cmp_fail = 1 or (LCM_STATUS.stpass = 0 and LCM_STATUS.stdone = 1) or (LCM_STATUS.cmp1_err_force_pass = 0 and LCM_STATUS.cmp1_err_force_done = 1) or (LCM_STATUS.cmp2_err_force_pass = 0 and LCM_STATUS.cmp2_err_force_done = 1) (v) LCM_STATUS.cmp1_err_force_done and LCM_STATUS.cmp1_err_force_pass flags need to be cleared before initiating the test a 2nd time. Reset type: SYSRSn
18-17	RESERVED	R	0h	Reserved
16	STEN	R-0/W	0h	0: configuration is ignored 1: self-test enabled (i) Self-test sequence will start when the bit is configured to a value of 1. The test shall be triggered only after enabling the lockstep feature. Lockstep feature shall not be disabled when the test is in progress. (ii) Once the test is initiated, both the comparators will be tested one after the other. It should be possible to execute this test with debugger connected (iii) The test can be triggered only after the previous execution of self-test is complete (i.e. ensuring by checking LCM_STATUS.stdone = 1) (iv) The test cannot be executed if there is pending functional failure or test failure (i.e. test cannot be executed when LCM_STATUS.cmp_fail = 1 or (LCM_STATUS.stpass = 0 and LCM_STATUS.stdone = 1) or (LCM_STATUS.cmp1_err_force_pass = 0 and LCM_STATUS.cmp1_err_force_done = 1) or (LCM_STATUS.cmp2_err_force_pass = 0 and LCM_STATUS.cmp2_err_force_done = 1) (v) LCM_STATUS.stdone and LCM_STATUS.stpass flags need to be cleared before initiating the test a 2nd time. (vi) Device shouldn't enter any low power modes when self-test is in progress Reset type: SYSRSn
15-1	RESERVED	R-0/W	0h	Reserved

Table 5-8. LCM_CONTROL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	CMPEN	R/W	0h	0: Lockstep compare disabled 1: Lockstep compare enabled Note: (1) Mentions of 'dual module' below are only applicable to systems that support dual modules. Datasheet will explicitly state this functionality if it exists. (2) The configuration to decide whether IP is in lockstep configuration, dual module configuration or single module configuration comes from system control. (3) This bit will have impact only when the IP is configured in lockstep mode (i.e. not in single module or dual module mode) (4) Device is expected to work in either lockstep mode or dual module mode. Switching between modes is not supported except the one time switching from lockstep mode to dual-core mode. (5) User must ensure that LCM_STATUS register should not indicate a failure (i.e. cmp_fail = 1, stpass = 0, cmp1_err_force_pass = 0, cmp2_err_force_pass = 0) at the time of enabling the lockstep compare. Reset type: SYSRSn

5.9.2.3 LCM_STATUS Register (Offset = 20h) [Reset = 00000001h]

LCM_STATUS is shown in Figure 5-5 and described in Table 5-9.

Return to the Summary Table.

LCM status register

Figure 5-5. LCM_STATUS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED	CMP2_ERR_FORCE_DONE	CMP2_ERR_FORCE_PASS	CMP1_ERR_FORCE_DONE	CMP1_ERR_FORCE_PASS	STACTIVE	STDONE	STPASS
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED							DBGCON
R-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED						CMP_FAIL	LSEN
R-0h						R-0h	R-1h

Table 5-9. LCM_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	0h	Reserved
22	CMP2_ERR_FORCE_DONE	R	0h	0: 'comparator2 compare error forcing test' in progress or not completed 1: 'comparator2 compare error forcing test' complete Note: If the bit is set, it need to be cleared before invoking the test 2nd time. Reset type: PORESETn
21	CMP2_ERR_FORCE_PASS	R	0h	0: 'comparator2 compare error forcing test' fail 1: 'comparator2 compare error forcing test' pass (comparator2 compare error output getting asserted during test is deemed as pass) Invoking this test will trigger an NMI on a test pass. Note: If the bit is set, it need to be cleared before invoking the test 2nd time. Reset type: PORESETn
20	CMP1_ERR_FORCE_DONE	R	0h	0: 'comparator1 compare error forcing test' in progress or not completed 1: 'comparator1 compare error forcing test' complete Note: If the bit is set, it need to be cleared before invoking the test 2nd time. Reset type: PORESETn
19	CMP1_ERR_FORCE_PASS	R	0h	0: 'comparator1 compare error forcing test' fail 1: 'comparator1 compare error forcing test' pass (comparator1 compare error output getting asserted during test is deemed as pass) Invoking this test will trigger an NMI on a test pass. Note: If the bit is set, it need to be cleared before invoking the test 2nd time. Reset type: PORESETn

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

Bit	Field	Type	Reset	Description
18	STACTIVE	R	0h	0: Self-test is not active 1: Self-test is active (in progress) The bit will be set in the next cycle of LCM_CONTROL.sten = 1 configuration and reset along with LCM_STATUS.stdone becoming '1'. Reset type: PORESETn
17	STDONE	R	0h	0: self-test in progress or not completed 1: self-test complete The bit will be zero by default and will become one once the self-test is completed. The test is deemed complete when the test sequence is complete or the test exits due to a failure. Note: If the bit is set, it need to be cleared before invoking the test 2nd time. Reset type: PORESETn
16	STPASS	R	0h	0: self-test fail 1: self-test pass The bit will be zero by default and will become one once the self-test is complete and status is pass Note: If the bit is set, it need to be cleared before invoking the self-test 2nd time. Reset type: PORESETn
15-9	RESERVED	R	0h	Reserved
8	DBGCON	R	0h	0: debugger is not connected 1: debugger is connected Note: The status is latched when debugger is connected. This can be cleared only by XRSn (a) When debugger is connected, lockstep comparison of the CPU is disabled. (b) Self-test can still be performed with debugger connected. (c) Error forcing mode cannot be checked with debugger connected Reset type: XRSn
7-2	RESERVED	R	0h	Reserved
1	CMP_FAIL	R	0h	0: Lockstep compare pass 1: Lockstep compare failed Note: (i) When the peripheral is configured to be in lockstep mode, the bit indicates whether lockstep comparison has failed. (i) Once the comparison is failed, Lockstep_compare_fail_status gets latched. It can be cleared only by a PORESETn or by writing to the status clear configuration. (iii) The bit will not get set during self-test mode or error forcing mode (iv) Self-test and compare error forcing check cannot be initiated when the cmp_fail flag value is 1'b1 Reset type: PORESETn
0	LSEN	R	1h	1: Peripheral is in lockstep configuration 0: peripheral is not in lockstep configuration. This configuration comes from system control. Note: lockstep_status is independent of the debugger connection. In order to check whether lockstep compare is disabled due to debugger connection, check LCM_STATUS.dbgcon Reset type: PORESETn

5.9.2.4 LCM_STATUS_CLEAR Register (Offset = 28h) [Reset = 00000000h]

LCM_STATUS_CLEAR is shown in Figure 5-6 and described in Table 5-10.

Return to the Summary Table.

LCM Status clear register

Figure 5-6. LCM_STATUS_CLEAR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED	CMP2_ERR_FORCE_DONE	CMP2_ERR_FORCE_PASS	CMP1_ERR_FORCE_DONE	CMP1_ERR_FORCE_PASS	RESERVED	STDONE	STPASS
R-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0h	R-0/W1C-0h	R-0/W1C-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						CMP_FAIL	RESERVED
R-0h						R-0/W1C-0h	R-0h

Table 5-10. LCM_STATUS_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	0h	Reserved
22	CMP2_ERR_FORCE_DONE	R-0/W1C	0h	0: No impact 1: LCM_STATUS.cmp2_err_force_done is reset to zero (If hardware is trying to set the flag and software is trying to clear the same flag in the same cycle, software clear is given higher priority) Reset type: PORESETn
21	CMP2_ERR_FORCE_PASS	R-0/W1C	0h	0: No impact 1: LCM_STATUS.cmp2_err_force_pass is reset to zero (If hardware is trying to set the flag and software is trying to clear the same flag in the same cycle, software clear is given higher priority) Reset type: PORESETn
20	CMP1_ERR_FORCE_DONE	R-0/W1C	0h	0: No impact 1: LCM_STATUS.cmp1_err_force_done is reset to zero (If hardware is trying to set the flag and software is trying to clear the same flag in the same cycle, software clear is given higher priority) Reset type: PORESETn
19	CMP1_ERR_FORCE_PASS	R-0/W1C	0h	0: No impact 1: LCM_STATUS.cmp1_err_force_pass is reset to zero (If hardware is trying to set the flag and software is trying to clear the same flag in the same cycle, software clear is given higher priority) Reset type: PORESETn
18	RESERVED	R	0h	Reset type: N/A
17	STDONE	R-0/W1C	0h	0: No impact 1: LCM_STATUS.stdone is reset to zero (If hardware is trying to set the flag and software is trying to clear the same flag in the same cycle, software clear is given higher priority) Reset type: PORESETn

Table 5-10. LCM_STATUS_CLEAR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	STPASS	R-0/W1C	0h	0: No impact 1: LCM_STATUS.stfail is reset to zero (If hardware is trying to set the flag and software is trying to clear the same flag in the same cycle, software clear is given higher priority) Reset type: PORESETn
15-2	RESERVED	R	0h	Reserved
1	CMP_FAIL	R-0/W1C	0h	0: No impact 1: LCM_STATUS.cmp_fail is reset to zero (If hardware is trying to set the flag and software is trying to clear the same flag in the same cycle, software clear is given higher priority) Reset type: PORESETn
0	RESERVED	R	0h	Reserved

5.9.2.5 PARITY_TEST Register (Offset = 68h) [Reset = 00000000h]

PARITY_TEST is shown in Figure 5-7 and described in Table 5-11.

Return to the Summary Table.

Enabling the parity test feature

Figure 5-7. PARITY_TEST Register

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

Table 5-11. PARITY_TEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	RESERVED	R	0h	Reserved
3-0	TESTEN	R/W	0h	1010: Parity test feature is enabled All other values: Parity test feature is disabled Note: (1) When the parity test feature is enabled, acutal registers are not accessible in the memory map. Instead, the parity values (final XOR output indicating the parity error) are accessible. Parity is computed for every byte and the correponding parity error value is available at the bit-0 of every byte. Value of '1' written to the parity bit after enabling the parity test feature can be used to inject the error by inverting the stored parity value. (2) It is recommended to leave the field as 0101 or 0000 after completing the parity test. Reset type: SYSRSn

5.9.2.6 LCM_LOCK Register (Offset = 70h) [Reset = 00000000h]

LCM_LOCK is shown in Figure 5-8 and described in Table 5-12.

Return to the Summary Table.

LCM lock configuration

Figure 5-8. LCM_LOCK Register

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

Table 5-12. LCM_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0	0h	Reserved
30	RESERVED	R-0	0h	Reserved
29	RESERVED	R-0	0h	Reserved
28	RESERVED	R-0	0h	Reserved
27	RESERVED	R-0	0h	Reserved
26	PARITY_TEST	R/W	0h	0: Register configuration is not locked. 1: Register configuration is locked. Reset type: SYSRSn
25	RESERVED	R-0	0h	Reserved
24	RESERVED	R-0	0h	Reserved
23	RESERVED	R-0	0h	Reserved
22	RESERVED	R-0	0h	Reserved
21	RESERVED	R-0	0h	Reserved
20	RESERVED	R-0	0h	Reserved
19	RESERVED	R-0	0h	Reserved
18	RESERVED	R-0	0h	Reserved
17	RESERVED	R-0	0h	Reserved
16	RESERVED	R-0	0h	Reserved
15	RESERVED	R-0	0h	Reserved
14	RESERVED	R-0	0h	Reserved
13	RESERVED	R-0	0h	Reserved
12	RESERVED	R-0	0h	Reserved
11	RESERVED	R-0	0h	Reserved

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

Bit	Field	Type	Reset	Description
10	LCM_STATUS_CLEAR	R/W	0h	0: Register configuration is not locked. 1: Register configuration is locked. Reset type: SYSRSn
9	RESERVED	R-0	0h	Reserved
8	RESERVED	R-0	0h	Reserved
7	RESERVED	R-0	0h	Reserved
6	RESERVED	R-0	0h	Reserved
5	RESERVED	R-0	0h	Reserved
4	RESERVED	R-0	0h	Reserved
3	RESERVED	R-0	0h	Reserved
2	LCM_CONTROL	R/W	0h	0: Register configuration is not locked. 1: Register configuration is locked. Reset type: SYSRSn
1	RESERVED	R-0	0h	Reserved
0	RESERVED	R-0	0h	Reserved

5.9.2.7 LCM_COMMIT Register (Offset = 78h) [Reset = 00000000h]

LCM_COMMIT is shown in Figure 5-9 and described in Table 5-13.

Return to the Summary Table.

LCM commit configuration

Figure 5-9. LCM_COMMIT Register

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

Table 5-13. LCM_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0	0h	Reserved
30	RESERVED	R-0	0h	Reserved
29	RESERVED	R-0	0h	Reserved
28	RESERVED	R-0	0h	Reserved
27	RESERVED	R-0	0h	Reserved
26	PARITY_TEST	R/WSonce	0h	0: Register lock configuration is not committed. 1: Register lock configuration is committed. Once configuration is committed, only reset can change the configuration. Reset type: SYSRSn
25	RESERVED	R-0	0h	Reserved
24	RESERVED	R-0	0h	Reserved
23	RESERVED	R-0	0h	Reserved
22	RESERVED	R-0	0h	Reserved
21	RESERVED	R-0	0h	Reserved
20	RESERVED	R-0	0h	Reserved
19	RESERVED	R-0	0h	Reserved
18	RESERVED	R-0	0h	Reserved
17	RESERVED	R-0	0h	Reserved
16	RESERVED	R-0	0h	Reserved
15	RESERVED	R-0	0h	Reserved
14	RESERVED	R-0	0h	Reserved
13	RESERVED	R-0	0h	Reserved
12	RESERVED	R-0	0h	Reserved
11	RESERVED	R-0	0h	Reserved

Table 5-13. LCM_COMMIT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	LCM_STATUS_CLEAR	R/WSonce	0h	0: Register lock configuration is not committed. 1: Register lock configuration is committed. Once configuration is committed, only reset can change the configuration. Reset type: SYSRSn
9	RESERVED	R-0	0h	Reserved
8	RESERVED	R-0	0h	Reserved
7	RESERVED	R-0	0h	Reserved
6	RESERVED	R-0	0h	Reserved
5	RESERVED	R-0	0h	Reserved
4	RESERVED	R-0	0h	Reserved
3	RESERVED	R-0	0h	Reserved
2	LCM_CONTROL	R/WSonce	0h	0: Register lock configuration is not committed. 1: Register lock configuration is committed. Once configuration is committed, only reset can change the configuration. Reset type: SYSRSn
1	RESERVED	R-0	0h	Reserved
0	RESERVED	R-0	0h	Reserved

Chapter 6

Peripheral Interrupt Priority and Expansion (PIPE)

This chapter describes the peripheral interrupt priority and expansion (PIPE) module. The PIPE module controls peripheral interrupt handling on the device.

6.1 Introduction

Each PIPE module instance arbitrates peripheral interrupts for the respective CPU. All asserted interrupts are arbitrated each clock cycle, with the highest priority interrupt asserted to the corresponding CPU interrupt line (NMI, RTINT, or INT). The PIPE module is responsible for providing vector addresses to the CPU for NMI, RTINT, INT and RESET. The PIPE is capable of custom ordering of interrupts, prioritization, and nesting.

6.1.1 Features

The PIPE module has the following features:

- Hardware support for interrupt prioritization, arbitration, grouping, software hand-shake, and nesting.
- Dynamic arbitration of interrupts in hardware on every clock.
- Selectable priority level to choose interrupts as either RTINT and INT.
- Interrupt grouping of adjacent-prioritized interrupts to block nesting within groups.
- Default index-based priority order for interrupts used in arbitration.
- Vector fetch support for RESET, NMI, RTINT, and INT.
- User access to the stack configured for INT.
- Contexts used by a software task manager system or operating system.
- Link-based protection verifies only legal code from the assigned interrupt owner services the interrupt.
- Device level protection validates only legal code source updates interrupt configuration and vector tables.
- Automatic context save and restore for RTINT and NMI.
- RTINT stack overflow protection always provides NMI a block of reserved stack space for execution.
- ECC protection for interrupt vector table.
- Parity protection for configuration registers.
- Optional locking capability of interrupt configurations.

6.1.2 Interrupt Concepts

An interrupt is a signal that causes the CPU to pause the currently running process and branch to a different piece of code known as an interrupt service routine (ISR). This is a useful mechanism for handling peripheral events, and involves less CPU overhead and program complexity than register polling. However, because interrupts are asynchronous to the program flow, care must be taken to avoid conflicts over resources that are accessed both in interrupts and in the main program code.

Interrupts propagate to the CPU through a series of flag and enable registers. The flag registers store the interrupt until the interrupt is processed. The enable registers allow or block the propagation of the interrupt. When an interrupt signal reaches the CPU, the CPU fetches the appropriate ISR address from the vector table.

6.1.3 PIPE Related Collateral

Foundational Materials

- C29x Academy - Interrupts

6.2 Interrupt Architecture

The PIPE module has three primary functional blocks:

1. Dynamic Priority Arbitration Circuit
2. Post Processing Block
3. Memory-Mapped Registers (includes vector table and bus interface)

These three blocks are explained in detail in the following subsections.

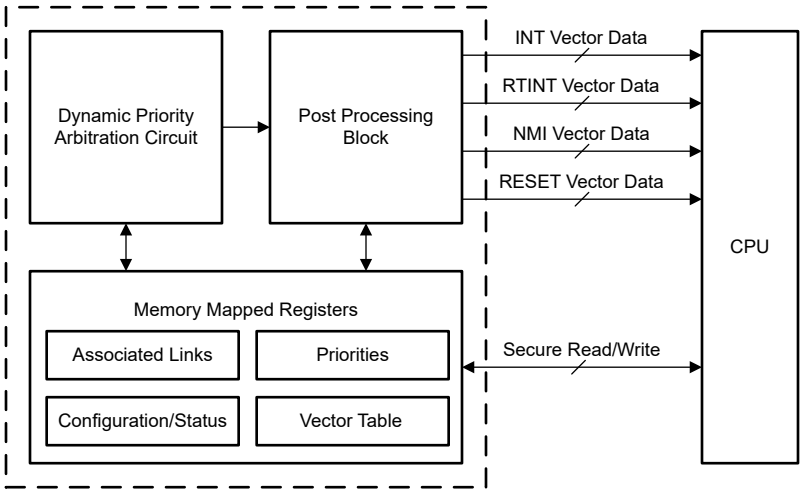


Figure 6-1. PIPE Architecture

6.2.1 Dynamic Priority Arbitration Block

The dynamic priority arbitration block provides the CPU with the highest priority interrupt vector that is available every clock cycle. The CPU processes the highest priority interrupt at the provided vector address.

6.2.2 Post Processing Block

The post processing block takes the highest priority interrupt that won the arbitration process and selects which interrupt line (INT or RTINT) to forward the interrupt to. The post processing block also automatically checks which link is accessing an interrupt line and whether a secure link is accessing protected registers.

Note

The NMI line provided to the CPU is an independent line that overrides any other interrupt (INT or RTINT) ready for assertion.

6.2.3 Memory-Mapped Registers

The memory-mapped registers (MMR) contain the interrupt configuration registers. Below are the type of registers available in the memory-mapped registers:

- Link associated with each interrupt.
- Priority configured for each interrupt.
- Interrupt configurations.
- Interrupt status.
- Vector table.

Accesses are controlled by the same security rules that apply to all registers.

6.3 Interrupt Propagation

Interrupts propagate to the CPU through several steps. Peripheral interrupts set the corresponding FLAG bit in the INT_CTL_REG_L_y register of a given interrupt. If the EN bit of the interrupt's INT_CTL_REG_L_y register is set, the interrupt propagates to the dynamic priority arbitration circuit. Next, the dynamic priority arbitration block and post processing block arbitrate the highest priority interrupt and assert this to the CPU on one of the two interrupt lines (RTINT or INT). Finally, the CPU chooses the highest priority interrupt line that is asserted (amongst NMI, RTINT, and INT) and begins execution of that interrupt.

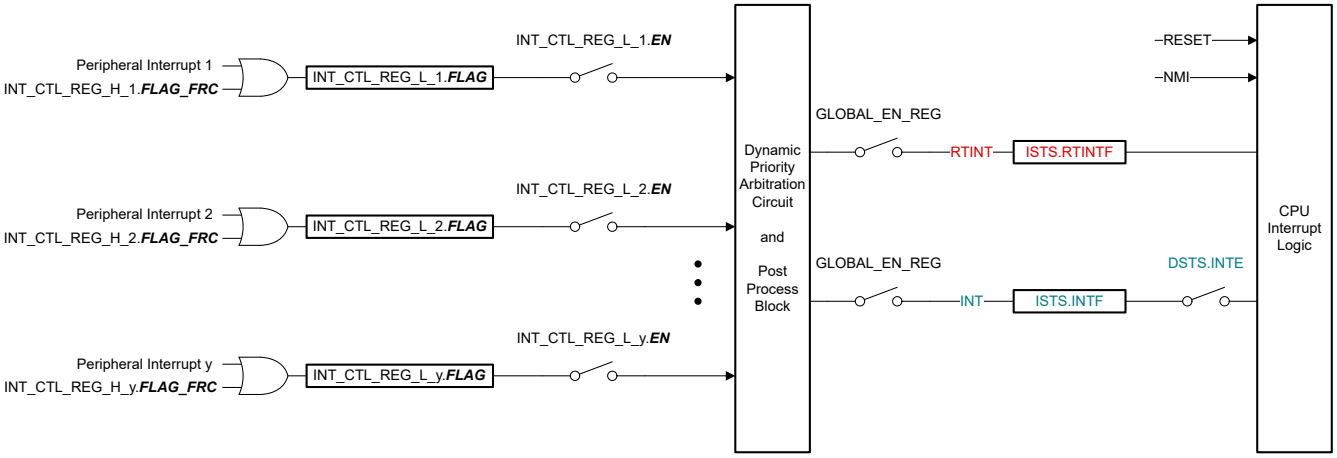


Figure 6-2. Interrupt Propagation

Note

NMI is an independent line that overrides any other interrupts ready for assertion. An NMI event asserted to PIPE is forwarded to the CPU, including NMI in service. CPU can select a new NMI at any time, but the PIPE module does not forward the new NMI until the in-service NMI interrupt service routine is complete.

The same rule is applicable to RESET. Once the CPU receives the RESET, there are no conditions to meet before reset assertion to CPU.

6.4 Configuring Interrupts

At power-up, all interrupts return to the default reset conditions:

- All interrupts are disabled.
- All interrupts are set to the default priority (INT_CONFIG.PRI_LEVEL).
- All interrupts are set to the default context (INT_CONFIG.CONTEXT_ID).
- All interrupts are assigned as INT by default (RTINT_THRESHOLD.INT_RTINT_THRESHOLD).
- All interrupt vectors point to the default NMI vector (INT_VECT_ADDR.*).
- Stack pointer warning levels are reset to default values (RTINT_SP_H.*).
- All interrupt owner and API links are set to a default link (INT_LINK_OWNER.*). The user needs to update the link to SROOT_LINK (LINK2) to be able to configure the PIPE even when SSU is in Mode 1.

Interrupts are configured as follows:

1. **Disable INT and RTINT:** Disable INT and RTINT using the GLOBAL_EN.ENABLE/KEY fields.
2. **Prioritization:** Set the interrupt priority as required.
3. **Nesting:** Set the group as required.
4. **Context:** Set the context ID as required.
5. **Stack protection:** Set stack pointer warning levels (RTINT_SP_H.*) as required.
6. **Enabling interrupts:** Enable required interrupts using INT_CTL_L_y.EN bit for each interrupt.
7. **Enabling INT and RTINT:** Enable INT and RTINT using the GLOBAL_EN.ENABLE/KEY bits.

Note

Enables/disables must be done within an ATOMIC block for predictable CPU behavior

Interrupts can be re-configured while GLOBAL_EN is enabled. However, interrupts that have already been forwarded from the PIPE to the CPU and have reached the CPU, being serviced, or are nested-in are not affected by any re-configuration changes. These settings do not apply until the next interrupt is forwarded from the PIPE to the CPU.

6.4.1 Enabling and Disabling Interrupts

Individual interrupts can be enabled first, using the INT_CTL_L.EN register bit. Once the desired interrupts are enabled at the peripheral level coming into PIPE, the user can then enable the PIPE module to begin forwarding interrupts to the CPU.

To enable the PIPE to forward INT and RTINT interrupts to the CPU, GLOBAL_EN.ENABLE/KEY bits must be set properly. See the register description for GLOBAL_EN register for values required for enabling the PIPE forwarding.

When disabling interrupts, caution must be taken to avoid unexpected interrupts from occurring. The steps below must be taken when disabling one or more interrupts.

1. Global disable PIPE interrupts (GLOBAL_EN.ENABLE).
2. Disable the individual interrupts desired (INT_CTL_L.EN).
3. Wait 6 cycles for any ongoing arbitration to complete.
4. Global enable PIPE interrupts (GLOBAL_EN disable).

Note

In this chapter, references to the global enable are referring to the GLOBAL_EN within the PIPE module which is different than the DSTS.INTE within the CPU that is needed to propagate INTs to the CPU.

6.4.2 Prioritization

The PIPE arbitrates incoming interrupts based on interrupt priority, with the lowest value being the highest priority. Priority is determined for each interrupt using two different values:

1. User-configured interrupt priority.
2. Index-based fixed interrupt priority.

6.4.2.1 User-Configured Interrupt Priority

User-configured interrupt priority is configured in software through the INT_CONFIG.PRI_LEVEL field that exists for every interrupt. The lower the PRI_LEVEL, the higher the priority of the interrupt. For example, an interrupt with PRI_LEVEL = 0 is serviced before an interrupt with PRI_LEVEL = 1 if both arrive at the same time.

When multiple interrupts are configured to the same PRI_LEVEL, the interrupts with the same PRI_LEVEL use the index-based fixed interrupt priority to determine the highest priority interrupt.

Note

By default every interrupt is configured to the highest PRI_LEVEL value (lowest priority). Therefore, all interrupts use index-based fixed interrupt priority by default.

6.4.2.2 Index-Based Fixed Interrupt Priority

The input number at which a particular interrupt is connected to the PIPE is called the "interrupt index". This is the index-based fixed interrupt priority that is fixed per device. This is also the indexing used to configure the interrupts in the PIPE.

For example, the INT_CONFIG register is repeated for every interrupt, and is indexed as INT_CONFIG_y (where y is the index of each interrupt).

Table 6-1 shows the "interrupt index" for all peripheral interrupt channels on this device.

Table 6-1. PIPE Channel Mapping

Vector Number	INT Signal Name
0	C29_CPU_OVFINT
1	C29_CPU_UVFINT
2	C29_CPU_DOVINT
3	WDINT
4	ERAD_INT
5	DLT_INT
6	CPU_TINT0
7	CPU_TINT1
8	CPU_TINT2
9	C29DBG2CPU_INT
10	ESM_CPUx_LOW_PRIORITY_INT
11	ESM_PARITYERRINT
12	IPC_INT1_1
13	IPC_INT1_2
14	IPC_INT1_3
15	IPC_INT1_4
16	IPC_INT2_1
17	IPC_INT2_2
18	IPC_INT2_3
19	IPC_INT2_4
20	IPC_HSM_RACK
21	IPC_HSM_WDONE
22-25	Reserved
26	HSM_CRYPTOENGINE_DTHE_TRNG_INT
27	HSM_CRYPTOENGINE_DTHE_PKAE_INT
28	HSM_CRYPTOENGINE_DTHE_SHA_S_INT
29	HSM_CRYPTOENGINE_DTHE_SHA_P_INT
30	HSM_CRYPTOENGINE_DTHE_AES_S_INT
31	HSM_CRYPTOENGINE_DTHE_AES_P_INT
32	HSM_CRYPTOENGINE_DTHE_SM3_INT
33	HSM_CRYPTOENGINE_DTHE_SM4_INT
34	FLC1_INT
35	FLC2_INT
36	EPWM1_INT
37	EPWM1_TZINT
38	EPWM2_INT
39	EPWM2_TZINT
40	EPWM3_INT
41	EPWM3_TZINT
42	EPWM4_INT
43	EPWM4_TZINT
44	EPWM5_INT
45	EPWM5_TZINT

Table 6-1. PIPE Channel Mapping (continued)

Vector Number	INT Signal Name
46	EPWM6_INT
47	EPWM6_TZINT
48	EPWM7_INT
49	EPWM7_TZINT
50	EPWM8_INT
51	EPWM8_TZINT
52	EPWM9_INT
53	EPWM9_TZINT
54	EPWM10_INT
55	EPWM10_TZINT
56	EPWM11_INT
57	EPWM11_TZINT
58	EPWM12_INT
59	EPWM12_TZINT
60	EPWM13_INT
61	EPWM13_TZINT
62	EPWM14_INT
63	EPWM14_TZINT
64	EPWM15_INT
65	EPWM15_TZINT
66	EPWM16_INT
67	EPWM16_TZINT
68	EPWM17_INT
69	EPWM17_TZINT
70	EPWM18_INT
71	EPWM18_TZINT
72	EQEP1_INT
73	EQEP2_INT
74	EQEP3_INT
75	EQEP4_INT
76	EQEP5_INT
77	EQEP6_INT
78	ECAP1_INT
79	ECAP2_INT
80	ECAP3_INT
81	ECAP4_INT
82	ECAP5_INT
83	HRCAP5_INT
84	ECAP6_INT
85	HRCAP6_INT
86	ADCA_EVT_INT
87	ADCAINT1
88	ADCAINT2
89	ADCAINT3
90	ADCAINT4

Table 6-1. PIPE Channel Mapping (continued)

Vector Number	INT Signal Name
91	ADCB_EVT_INT
92	ADCBINT1
93	ADCBINT2
94	ADCBINT3
95	ADCBINT4
96	ADCC_EVT_INT
97	ADCCINT1
98	ADCCINT2
99	ADCCINT3
100	ADCCINT4
101	ADCD_EVT_INT
102	ADCDINT1
103	ADCDINT2
104	ADCDINT3
105	ADCDINT4
106	ADCE_EVT_INT
107	ADCEINT1
108	ADCEINT2
109	ADCEINT3
110	ADCEINT4
111	Reserved
112	SD1_ERRINT
113	SD1FLT1_DRINT
114	SD1FLT2_DRINT
115	SD1FLT3_DRINT
116	SD1FLT4_DRINT
117	SD2_ERRINT
118	SD2FLT1_DRINT
119	SD2FLT2_DRINT
120	SD2FLT3_DRINT
121	SD2FLT4_DRINT
122	SD3_ERRINT
123	SD3FLT1_DRINT
124	SD3FLT2_DRINT
125	SD3FLT3_DRINT
126	SD3FLT4_DRINT
127	SD4_ERRINT
128	SD4FLT1_DRINT
129	SD4FLT2_DRINT
130	SD4FLT3_DRINT
131	SD4FLT4_DRINT
132	CLB1_INT
133	CLB2_INT
134	CLB3_INT
135	CLB4_INT

Table 6-1. PIPE Channel Mapping (continued)

Vector Number	INT Signal Name
136	CLB5_INT
137	CLB6_INT
138	UARTA_INT
139	UARTB_INT
140	UARTC_INT
141	UARTD_INT
142	UARTE_INT
143	UARTF_INT
144	LINA_0
145	LINA_1
146	LINB_0
147	LINB_1
148	PMBUSA_INT
149	I2CA_INT
150	I2CA_FIFO
151	I2CB_INT
152	I2CB_FIFO
153	SPIA_TXINT
154	SPIA_RXINT
155	SPIB_TXINT
156	SPIB_RXINT
157	SPIC_TXINT
158	SPIC_RXINT
159	SPID_TXINT
160	SPID_RXINT
161	SPIE_TXINT
162	SPIE_RXINT
163	FSITXA_INT1
164	FSITXA_INT2
165	FSITXB_INT1
166	FSITXB_INT2
167	FSITXC_INT1
168	FSITXC_INT2
169	FSITXD_INT1
170	FSITXD_INT2
171	FSIRXA_INT1
172	FSIRXA_INT2
173	FSIRXB_INT1
174	FSIRXB_INT2
175	FSIRXC_INT1
176	FSIRXC_INT2
177	FSIRXD_INT1
178	FSIRXD_INT2
179	SENT1_INT
180	SENT2_INT

Table 6-1. PIPE Channel Mapping (continued)

Vector Number	INT Signal Name
181	SENT3_INT
182	SENT4_INT
183	SENT5_INT
184	SENT6_INT
185	Reserved
186	MCANA_WAKE_AND_TS_PLS_INT
187	MCANA_INT1
188	MCANA_INT0
189	Reserved
190	MCANB_WAKE_AND_TS_PLS_INT
191	MCANB_INT1
192	MCANB_INT0
193	Reserved
194	MCANC_WAKE_AND_TS_PLS_INT
195	MCANC_INT1
196	MCANC_INT0
197	Reserved
198	MCAND_WAKE_AND_TS_PLS_INT
199	MCAND_INT1
200	MCAND_INT0
201	Reserved
202	MCANE_WAKE_AND_TS_PLS_INT
203	MCANE_INT1
204	MCANE_INT0
205	Reserved
206	MCANF_WAKE_AND_TS_PLS_INT
207	MCANF_INT1
208	MCANF_INT0
209	ECAT_INTn
210	ECAT_SYNC0
211	ECAT_SYNC1
212	ECAT_RST
213	RTDMA1_CH1INT
214	RTDMA1_CH2INT
215	RTDMA1_CH3INT
216	RTDMA1_CH4INT
217	RTDMA1_CH5INT
218	RTDMA1_CH6INT
219	RTDMA1_CH7INT
220	RTDMA1_CH8INT
221	RTDMA1_CH9INT
222	RTDMA1_CH10INT
223	RTDMA2_CH1INT
224	RTDMA2_CH2INT
225	RTDMA2_CH3INT

Table 6-1. PIPE Channel Mapping (continued)

Vector Number	INT Signal Name
226	RTDMA2_CH4INT
227	RTDMA2_CH5INT
228	RTDMA2_CH6INT
229	RTDMA2_CH7INT
230	RTDMA2_CH8INT
231	RTDMA2_CH9INT
232	RTDMA2_CH10INT
233	WADI1_INTN_O
234	WADI2_INTN_O
235	XINT1
236	XINT2
237	XINT3
238	XINT4
239	XINT5
240	DCC1_DONE
241	DCC2_DONE
242	DCC3_DONE
243	LPM_WAKEINT
244	SWINT12
245	SWINT11
246	SWINT10
247	SWINT9
248	SWINT8
249	SWINT7
250	SWINT6
251	SWINT5
252	SWINT4
253	SWINT3
254	SWINT2
255	SWINT1

This recommended interrupt handling process can be followed for level interrupts to avoid unnecessary interrupts getting triggered:

1. Enter ISR.
2. Perform ISR functions.
3. Clear interrupt status within peripheral.
4. Wait for peripheral interrupt status to get cleared using NOPs or poll for interrupt flag status to get cleared.
5. Write to INT_{#}_CTL_REG_H.FLAG_CLR to clear the INT_{#}_CTL_REG_L.FLAG in software.
6. Wait for the PIPE flag to get cleared by writing 7 NOPs.
7. Exit ISR.
8. Clear interrupt within peripheral.

6.4.3 Nesting and Priority Grouping

Interrupt Nesting

By default, nesting is supported at the hardware level in the C29x CPU. The PIPE module allows for interrupt types RTINT and INT to nest within one another. A higher priority interrupt nests within a lower priority interrupt. Nesting for INTs within the PIPE module is enabled within a Interrupt Service Routine (ISR) by setting the CPU level DSTS.INTE bit active because this bit is disabled while entering the ISR.

Grouping Interrupts Based on Priority Levels

The PIPE also allows for interrupts to prevent nesting within other interrupts using "groups". Interrupts in a given group do not preempt or nest within other interrupts in the group.

To form a group, the least significant bits of the priority (INT_CONFIG.PRI_LEVEL) are used to choose which interrupts are in the same group. The number of least significant bits of the INT_CONFIG.PRI_LEVEL that is used for grouping interrupts is also configurable (masked using the INT_GRP_MASK register) in binary multiples (ex: 2, 4, 8, 16). Groups are made up of interrupts that are near one another in configured priority levels.

Note

The RTINT_THRESHOLD needs to be aligned with the configured group mask to maintain proper RTINT behavior. The PIPE module first groups the interrupts and then processes them to be RTINT or INT. In other words, grouping takes priority over RTINT resolution based on the threshold value.

For example when this above note is not followed, some group includes both priority levels INT_7 and INT_29 and the RTINT_THRESHOLD is set to 10. If the CPU was servicing INT_29 (INT) and INT_7 comes (RTINT), irrespective of the RTINT threshold, INT_7 is not forwarded on RTINT/INT lines.

The group mask combinations possible are outlined in Table 6-2. Any other group mask chosen for INT_GRP_MASK that is not listed in Table 6-2 results in using the default group mask of 0xFF where all interrupts can nest.

Table 6-2. Group Mask Combinations

Group Mask	Group Size	Number of Groups	Priority Assignment Distance	Comment
0xFF/Other	1	256	1	0xFF: All interrupts can nest in one another. All interrupt levels are in individual groups, so 256 configurable priorities are available with 1 interrupt level in each. All other values: If any other value is used, the mask value defaults to 0xFF that correlates to no group mask.
0xFE	2	128	2	128 priority groups with 2 interrupt levels in each; hence, the number of priorities reduce to 128 for preemption.
0xFC	4	64	4	64 priority groups with 4 interrupt levels in each; hence, the number of priorities reduce to 64 for preemption.
0xF8	8	32	8	32 priority groups with 8 interrupt levels in each; hence, the number of priorities reduce to 32 for preemption.
0xF0	16	16	16	16 priority groups with 16 interrupt levels in each; hence, the number of priorities reduce to 16 for preemption.
0xE0	32	8	32	8 priority groups with 32 interrupt levels in each; hence, the number of priorities reduce to 8 for preemption.
0xC0	64	4	64	4 priority groups with 64 interrupt levels in each; hence, the number of priorities reduce to 4 for preemption.
0x80	128	2	128	2 priority groups with 128 interrupt levels in each; hence, the number of priorities reduce to 2 for preemption.

Table 6-2. Group Mask Combinations (continued)

Group Mask	Group Size	Number of Groups	Priority Assignment Distance	Comment
0x00	256	1	256	No interrupts can nest in one another. All interrupts are in one group, so configurable priorities are used for the highest priority interrupt. There is no nesting of interrupt priority levels regardless of INT or RTINT status.

For example, in the case of an interrupt with priority level 13, this interrupt can belong to different groups depending on the group mask: 0xFF: Group 14, 0xFC: Group 11, 0xFC: Group 4, and so on.

Figure 6-3 illustrates one example of how grouping allows interrupts to nest, and is shown with PRI_LEVEL 0 to 7 set and group sizes selected as shown. In this example, no two interrupts have the same interrupt priority level configurations.

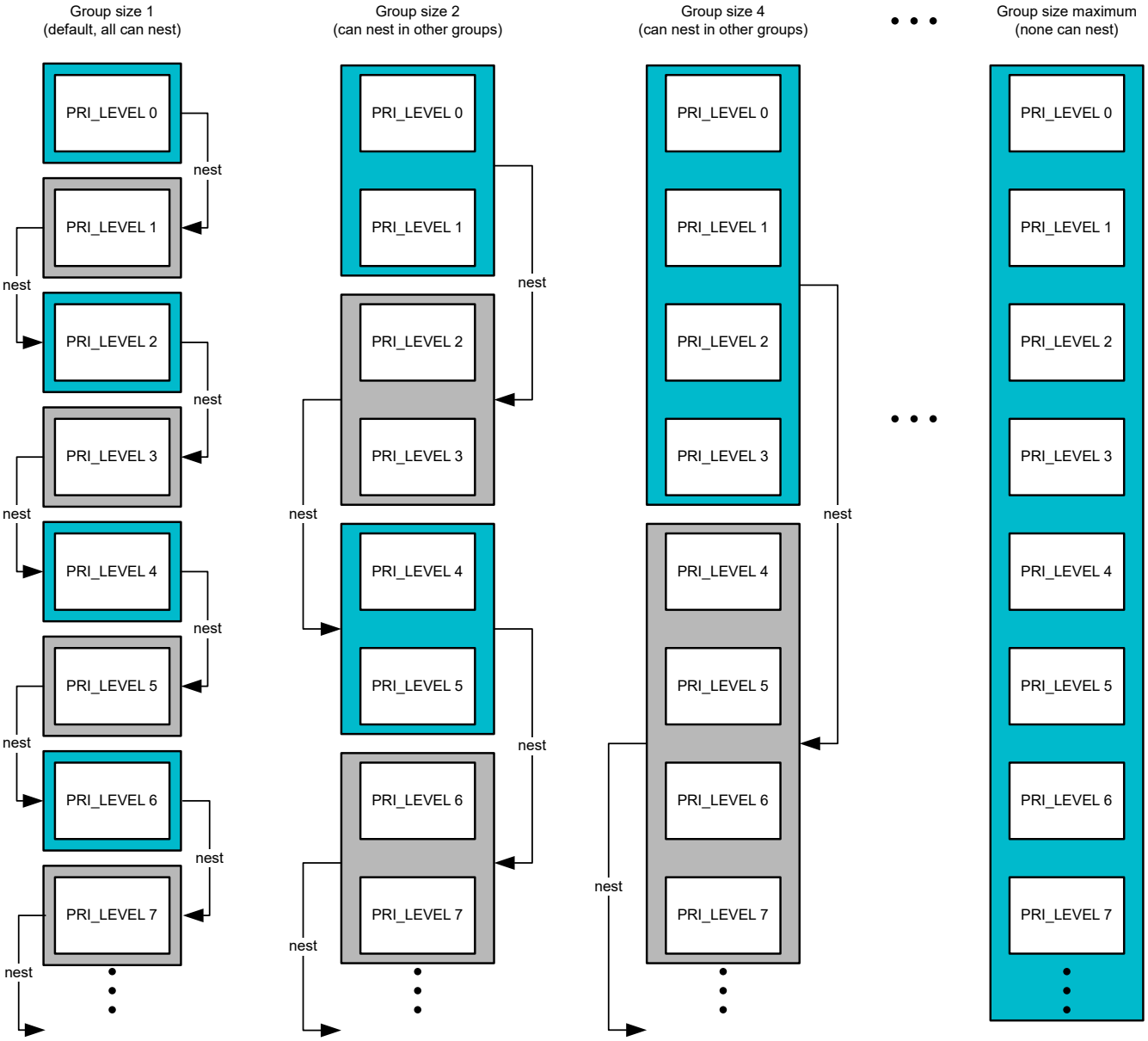


Figure 6-3. Interrupt Grouping

6.4.4 Stack Protection

INT uses the software stack that is configured in the INTSP register. The INT stack has no hardware stack overflow protection. Software can be used to implement a software stack overflow protection scheme for the INT stack as required.

RTINT and NMI use the RTINT stack which is a dedicated hardware stack used for RTINT and NMI context save and restore.

Context save for RTINT and NMI executes the following in hardware automatically upon ISR entry:

- Increment RTISP (interrupt stack pointer for RTINT stack).
- Transfer all CPU registers to RTINT stack.
- Clear CPU registers before ISR start.

Context restore for RTINT and NMI executes the following in hardware automatically upon ISR exit:

- Copy all CPU registers back from RTINT stack.
- Decrement RTISP.

To avoid excessive nesting of the RTINT stack leading to stack overflow, the CPU tracks the level of nesting on RTINT and outputs this information to the PIPE.

WARNRTISP

The first line of defense against stack overflow is the user-configurable RTINT_SP_H.WARNRTISP register field. The number of stack blocks used by the RTINT stack (counted by RTISP) is compared to the level configured in RTINT_SP_H.WARNRTISP. If RTISP is greater than or equal to RTINT_SP_H.WARNRTISP, the PIPE changes which interrupts are allowed to be asserted: only interrupts with a priority level higher than the user-configured RTINT_SP_H.WARNRTISP_PRIO_LEVEL are allowed. This can be summarized using the pseudo-code below.

```
if(RTISP >= RTINT_SP_H.WARNRTISP){
    Only allow RTINT with INT_CONFIG.PRI_LEVEL < RTINT_SP_H.WARNRTISP_PRIO_LEVEL
}
else{
    Allow all RTINT.
}
```

The status of whether or not RTISP level has reached the user-configured WARNRTISP level is located in the RTISP_STS.WARNRTISP_STS field.

Note

WARNRTISP must not be configured as 0 to avoid immediately firing an error on a register write.

WARNRTISP must not be configured as 1 to avoid firing an error to the ESM on the first RTINT occurrence. However, there is no impact if the user is not utilizing and/or expecting any RTINT.

MAXRTISP

When RTISP = MAXRTISP, all RTINT stack space has been exhausted except for one stack slot which is used by a NMI to alert the CPU of the overflow. When this condition is true, the PIPE does the following:

- Stops asserting any and all new RTINTs to the CPU.
- Raises a fault to the Error Signaling Module (ESM).
- Sets this max breach status in RTISP_STS.MAXRTISP_STS.

The MAXRTISP puts the CPU in a fault state, and the ESM responds to the PIPE fault assertion by raising an NMI back to the PIPE. However, PIPE does not allow any other NMIs to nest within this NMI. PIPE only allows further RTINTs to execute once the second-to-last stack slot is available again in the RTINT stack. A system reset is required to recover the device from fault.

6.4.5 Context

PIPE allows interrupts to be grouped into "contexts". These contexts can be used by a software task manager system or operating system to switch between multiple execution contexts.

Each interrupt is configured with a CONTEXT_ID (INT_CONFIG.CONTEXT_ID) that determines which context the interrupt is in. By default, all interrupts are assigned the same CONTEXT_ID.

The PIPE can then be configured to only arbitrate interrupts whose context matches the "active context". Active context is selected by setting the TASK_CTRL.ACTIVE_CONTEXT_ID field.

Any updates to INT_CONFIG.CONTEXT_ID and TASK_CTRL.ACTIVE_CONTEXT_ID take place on the next arbitration cycle.

Changing the active context or context ID does not stop accumulation of interrupt events that are out of context. This only disables interrupt participation in arbitration.

Note

GLOBAL_EN.ENABLE must be disabled before changing TASK_CTRL.ACTIVE_CONTEXT_ID. This is done to avoid disrupting the priority arbitration as some interrupts can become disabled and out of arbitration and others can be enabled and join arbitration.

Once TASK_CTRL.ACTIVE_CONTEXT_ID is modified, software must wait at least 6 system clock cycles for arbitration to complete before re-enabling GLOBAL_EN.ENABLE.

6.5 Safety and Security

The PIPE has protection mechanisms to provide safety and security for the interrupt architecture.

6.5.1 Access Control

Access privileges for vector table and registers are explained in Table 6-3.

Table 6-3. Access Privileges

Access Privilege	Accessing Link	Reason	Check if PIPE matches CPU settings
OWNER_LINK	Owner link of interrupt line	Only the OWNER_LINK of the particular interrupt line is allowed to update or read the operational registers such as flag RD/SET/CLR, status RD, and so on.	CURRLINK=OWNER_LINK
API_LINK	API link of interrupt line when called by OWNER_LINK.	Common-code link access is allowed when API_LINK is enabled. This is needed as often times, the codes to update the PIPE in ISR are based on common-code link used across different interrupts.	CURRLINK=API_LINK && API_LINK_EN='1' && CLINK=OWNER_LINK
SROOT_LINK	Secure Root Link (LINK2)	SROOT_LINK is the most trusted secure code with access to all the PIPE registers to configure interrupts and handle exceptions beyond the regular ISR.	CURRLINK=SROOT_LINK
BOOT_LINK	Boot link (LINK1) access for interrupt handling during boot and boot loading.	BOOT_LINK needs access to configure PIPE and handle peripheral interrupts during boot process. As user code has not started yet, this access is for a short period. Access is granted as long the BOOT_LINK_LOCK is not set. Once the boot process is over, either the BOOT_LINK or SROOT_LINK shall update the BOOT_LINK_LOCK, disabling access.	CURRLINK=BOOT_LINK && BOOT_LINK_LOCK = '0'

6.5.2 PIPE Errors

The PIPE module is directly connected to the Error Aggregator which aggregates errors occurred during interrupt related errors such as interrupt service routines (ISRs). PIPE outputs the error, error type, and error address to the Error Aggregator. The errors from the CPUx and associated PIPE instance are combined as CPUx INT.

The errors are propagated to the ESM, PIPE, and CPU to take the appropriate action, typically a NMI. The following table summarizes various types of errors that can occur with INT, RTINT, NMI, and overall PIPE module.

Table 6-4. CPUx INT Error Aggregator Errors

CPUx INT Interface Errors	Priority	Description
MAIN/INT/RTINT/NMI ISR ENTRY ERROR	HIGH	Error in entering an interrupt service routine for a specific interrupt type.
MAIN/INT/RTINT/NMI CORRECTABLE VECTOR ERROR	LOW	Reading interrupt vector and/or link register related to a specific interrupt type that result in a single-bit ECC error.
MAIN/INT/RTINT/NMI UNCORRECTABLE VECTOR ERROR	HIGH	Reading interrupt vector address related to a specific interrupt type that result in a double-bit ECC error.
MAIN/INT/RTINT/NMI INTERRUPT RETURN ERROR	HIGH	Error in exiting an interrupt service routine for a specific interrupt type.
RTINT/NMI CONTEXT RESTORE CORRECTABLE VECTOR ERROR	LOW	CPU performs an ECC check on the vector address related to a specific interrupt type that result in a single-bit ECC error.
RTINT/NMI CONTEXT RESTORE UNCORRECTABLE VECTOR ERROR	HIGH	CPU performs an ECC check on the vector address related to a specific interrupt type that result in a double-bit ECC error.

Table 6-4. CPUx INT Error Aggregator Errors (continued)

CPUx INT Interface Errors	Priority	Description
NMI MAXISP ERROR	HIGH	CPU current interrupt stack pointer reaches the maximum allotted RTINT stack space (Ex. 15 for F29H85x).
PIPE SCRUBBING / VECTOR INIT CORRECTABLE ERROR	LOW	PIPE reads interrupt vector address that results in a single-bit ECC error.
PIPE SCRUBBING / VECTOR INIT UNCORRECTABLE ERROR	HIGH	PIPE reads interrupt vector and/or link register that results in a double-bit ECC error.
PIPE WARNISP	LOW	CPU current interrupt stack reaches or exceeds the value of the WARNISP Threshold. NOTE: This error is only automatically cleared when the interrupt stack pointer value falls below this threshold.
PIPE MAXISP	HIGH	CPU current interrupt stack pointer reaches the maximum allotted RTINT stack space (Ex. 15 for F29H85x).
PIPE SECURITY VIOLATION	HIGH	Unsecure access is made by illegitimate code during a PIPE register read and/or write.
PIPE REG PARITY ERROR	HIGH	Parity mismatch in a PIPE register and the respective stored parity value.
PIPE REGISTER PARITY DIAG ERROR	LOW	To determine if the parity checkers in the diagnostics registers are working correctly, user injects an error intentionally into this parity checker to see if an error is being generated as expected.
PIPE LOCK_KEY ERROR	LOW	Locked register is written to or a wrong key is used for a keyed register.

Note

Refer to Chapter 8and Chapter 7 for the complete list of error sources and information on how to read and use these errors for debugging purposes.

6.5.3 Register Data Integrity and Safety

The PIPE vector table and critical configuration registers have ECC protection. All other PIPE configuration registers have parity protection. The registers with parity-protection are mentioned in the registers section of this chapter.

6.5.4 Self-Test and Diagnostics

For diagnostics of safety mechanisms, ECC logic needs to be checked periodically. However, running test-of-diagnostics in software to check ECC logic is time consuming as many patterns have to be run to get to the required coverage. To enable seamless diagnostics, self-test logic is added.

The self-test controller generates test sequences to detect faults in ECC logic. Any errors found as part of the test sequences is logged into self-test configuration registers that are part of the PIPE module. Different data patterns can be required to get additional coverage. The input data and ECC values to the ECC logic is programmable through self-test registers.

If at any point the software reads the INT_VECT_ADDR registers to perform ECC checks on the read vector address value, any enabled self-test sequences are paused. Software is required to continually poll for the completion status of the self-test sequences to account for such pauses.

The self-test controller can run the following test sequences, see Figure 6-4, to detect an error in ECC logic:

1. Positive check: Input pattern (Data, Address, ECC bits) that is written does not result in ECC errors. No errors are generated by the ECC logic when none are expected.
2. 1-bit error check: Single bit errors are induced sequentially and then checked for the expected correctable error.
3. 2-bit error check: Double bit errors are induced sequentially and then checked for the expected uncorrectable error.

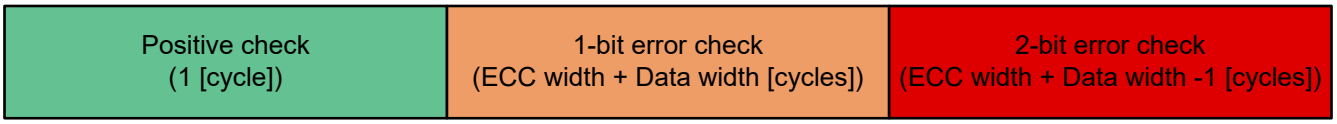


Figure 6-4. ECC Self-Test Sequence

If any of these checks fails, then self-test execution is aborted. The SELFTEST_DIAG_STATUS register can be read to check for these errors.

Programming sequence for self-test controller:

1. Enable self-test controller (SELFTEST_DIAG_CONTROL. DIAG_TEST_EN = 0x11).
2. Write the input data pattern to be used for self-test
   a. Write input pattern to SELFTEST_DIAG_DATAx registers with a data width of 52 bits.
3. Write the ECC pattern to SELFTEST_DIAG_ECC.
4. Execute ECCSELFTEST instruction.
5. Check for errors. If errors are found, take corrective action:
   a. SELFTEST_DIAG_STATUS.DIAG_FAIL_CHECK_TYPE can be read to determine the sequence that was failed (Positive check, Correctable error check, or Uncorrectable error check).
   b. SELFTEST_DIAG_STATUS. DIAG_FAIL_BIT_INDEX can be read to determine the data bit that was failed with error injection.
6. Clear the self-test status by writing to SELFTEST_DIAG_STATUS_CLR.

Refer to PIPE Self-Test registers for additional details.

6.6 Software

6.6.1 PIPE Registers to Driverlib Functions

Table 6-5. PIPE Registers to Driverlib Functions

File	Driverlib Function
RTINT_THRESHOLD	
interrupt.h	Interrupt_setThreshold
INT_GRP_MASK	
interrupt.h	Interrupt_setGroupMask
GLOBAL_EN	
interrupt.h	Interrupt_enableGlobal
interrupt.h	Interrupt_disableGlobal
REVISION	
-	
CPU_INT_STS	
-	
RST_VECT	
-	
RST_LINK_OWNER	
-	
NMI_STS	
-	
NMI_VECT	
interrupt.c	Interrupt_initVectorTable
interrupt.c	Interrupt_configNMI
interrupt.h	Interrupt_register
interrupt.h	Interrupt_unregister
NMI_LINK_OWNER	
interrupt.c	Interrupt_configNMI
interrupt.h	Interrupt_setLinkOwner
MEM_ECC_DIAG	
-	
MEM_INIT	
interrupt.c	Interrupt_initModule
MEM_INIT_STS	
interrupt.c	Interrupt_initModule
INT_SEC_STS	
-	
INT_SEC_CLR	
-	
RTINT_SP_L	
-	
RTINT_SP_H	
interrupt.h	Interrupt_setRTINTSPWarning
RTISP_STS	
-	
INTSP	

Table 6-5. PIPE Registers to Driverlib Functions (continued)

File	Driverlib Function
interrupt.h	Interrupt_setINTSP
LOCK	
interrupt.h	Interrupt_lockRegister
interrupt.h	Interrupt_unlockRegister
COMMIT	
interrupt.h	Interrupt_commitRegister
TASK_CTRL	
interrupt.h	Interrupt_setActiveContextID
interrupt.h	Interrupt_enableSupervisorIgnoreINTE
interrupt.h	Interrupt_disableSupervisorIgnoreINTE
BOOT_LINK_CTRL	
interrupt.h	Interrupt_lockBootLinkUpdates
INT_VECT_MAPPING	
-	
MMR_CLR	
interrupt.c	Interrupt_initModule
ALL_FLAG_CLR	
-	
REG_PARITY_DIAG_DATA	
-	
REG_PARITY_DIAG_PARITY	
-	
REG_PARITY_DIAG_ASSERT	
-	
REG_PARITY_CHECK	
-	
REG_PARITY_READ	
-	
INT_CTL_L(i)	
interrupt.c	Interrupt_configChannel
interrupt.h	Interrupt_enable
interrupt.h	Interrupt_disable
INT_CTL_H(i)	
interrupt.h	Interrupt_force
interrupt.h	Interrupt_clearFlag
interrupt.h	Interrupt_clearOverflowFlag
INT_CONFIG(i)	
interrupt.c	Interrupt_configChannel
interrupt.h	Interrupt_setPriority
interrupt.h	Interrupt_setContextID
INT_LINK_OWNER(i)	
interrupt.c	Interrupt_configChannel
interrupt.h	Interrupt_setLinkOwner
interrupt.h	Interrupt_setAPILink
INT_VECT_ADDR(i)	

Table 6-5. PIPE Registers to Driverlib Functions (continued)

File	Driverlib Function
interrupt.c	Interrupt_initVectorTable
interrupt.c	Interrupt_configChannel
interrupt.h	Interrupt_register
interrupt.h	Interrupt_unregister
INT_LINK_OWNER_LFU(i)	
-	
INT_VECT_ADDR_LFU(i)	
-	
SELFTEST_DIAG_DATA0	
-	
SELFTEST_DIAG_DATA1	
-	
SELFTEST_DIAG_ECC	
-	
SELFTEST_DIAG_CONTROL	
-	
SELFTEST_DIAG_STATUS	
-	
SELFTEST_DIAG_STATUS_CLR	
-	

6.6.2 INTERRUPT Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/interrupt

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

6.6.2.1 RTINT vs INT Latency example - SINGLE_CORE

FILE: interrupt_ex1_int_rtint_latency.c

This example demonstrates the interrupt latencies of INT and RTINT. This configures 2 CPU timer interrupts with following priorities.

- RTINT Threshold = 15
- Timer 0 Interrupt priority = 10 -> RTINT
- Timer 1 Interrupt priority = 20 -> INT

User can set the macro ENABLE_FUNCTION_CALL_IN_ISR to 1 to configure both the ISRs to invoke a function call, hence multiple core registers needs to be saved and restored. If the macro is set to 0, both ISRs just updates a counter variable, hence requires only 1 register to be saved and restored. In case of RTINT, register saves and restores are done by the hardware. In case of INT, register saves and restores are done by the software.

The ERAD module is configured to measure the cycles taken between the timer interrupt event to the first instruction in the ISR.

Note : Run the example for some duration to get the min and max values. The watch variables are arrays with min and max values [Min, Max]

External Connections

- None

Watch Variables

- timer0_intEvent_to_isr : Min and Max cycles count between the timer event to the ISR entry
- timer0_isr_to_data : Min and Max cycle count between the ISR entry to the first C instruction (data access to cpuTimer0IntCount variable)
- timer1_intEvent_to_isr : Min and Max cycles count between the timer event to the ISR entry
- timer1_isr_to_data : Min and Max cycle count between the ISR entry to the first C instruction (data access to cpuTimer1IntCount variable)

Expected results : (With opt level = 1)

| RTINT | RTINT | INT | INT |

| w/o func | with func | w/o func | with func |

intEvent_to_isr | 21-27 | 21-27 | 19-25 | 19-25 |

isr_to_data | 1 | 1 | 2 | 25-26 |

6.6.2.2 INT and RTINT Nesting Example - SINGLE_CORE

FILE: interrupt_ex2_int_rtint_nesting.c

This example showcases nesting of INTs and RTINTs in groups using software interrupts and increments every time the software asserts an interrupt. The watch variables and print statements can be used to determine entry and exit points of ISR.

The interrupt priorities / group mask are configured as follows :

- RTINT Threshold = 20
- Group Mask = 0xFE
- INT 1 Interrupt priority = 50 -> INT (Group 24)
- INT 2 Interrupt priority = 51 -> INT (Group 24)
- INT 3 Interrupt priority = 49 -> INT (Group 23)
- RTINT 1 Interrupt priority = 10 -> RTINT (Group 4)
- RTINT 2 Interrupt priority = 11 -> RTINT (Group 4)
- RTINT 3 Interrupt priority = 9 -> RTINT (Group 3)

Order of Interrupt Triggers: INT1 -> INT2 -> INT3 -> RTINT1 -> RTINT2 -> RTINT3

Order of Interrupt Execution: INT1 -> INT3 -> RTINT1 -> RTINT3 -> RTINT1 -> RTINT2 -> INT3 -> INT1 -> INT2

Sysconfig inserts the required attributes to the ISR functions to inform the compiler that the function is an interrupt / realtime interrupt.

External Connections

- None

Watch Variables

- *SWINT1Count* - INT1 count
- *SWINT2Count* - INT2 count
- *SWINT3Count* - INT3 count
- *SWINT4Count* - RTINT1 count
- *SWINT5Count* - RTINT2 count
- *SWINT6Count* - RTINT3 count

6.7 PIPE Registers

This section describes the PIPE Registers.

6.7.1 PIPE Base Address Table

Table 6-6. PIPE Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
PIPE_REGS	PIPE_BASE	0x3002_0000	-	YES	YES	YES	-	-	-	YES

6.7.2 PIPE_REGS Registers

Table 6-7 lists the memory-mapped registers for the PIPE_REGS registers. All register offset addresses not listed in Table 6-7 should be considered as reserved locations and the register contents should not be modified.

Table 6-7. PIPE_REGS Registers

Offset	Acronym	Register Name	Protection
0h	RTINT_THRESHOLD	Interrupt threshold register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, LOCK: LOCK.GLOBAL_LOCK,, PARITY
4h	INT_GRP_MASK	Interrupt group mask register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, LOCK: LOCK.GLOBAL_LOCK, PARITY
8h	GLOBAL_EN	Global enable for INT and RTINT	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, LOCK: LOCK.GLOBAL_LOCK,KEY:KEY=0xface, PARITY
Ch	REVISION	Reserved	PARITY
20h	CPU_INT_STS	CPU interrupt status	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, PARITY
24h	RST_VECT	Reset vector register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, LOCK: LOCK.GLOBAL_LOCK, PARITY
28h	RST_LINK_OWNER	Reset link owner register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, PARITY
2Ch	NMI_STS	Non Maskable Interrupt status register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, PARITY
30h	NMI_VECT	Non Maskable Interrupt vector register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, LOCK: LOCK.GLOBAL_LOCK, PARITY

Table 6-7. PIPE_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
34h	NMI_LINK_OWNER	Non Maskable Interrupt link owner register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, PARITY
40h	MEM_ECC_DIAG	ECC diagnostics register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, KEY:KEY=0x5a5a, PARITY
44h	MEM_INIT	PIPE vector memory initialization register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, KEY:KEY=0x5a5a
48h	MEM_INIT_STS	PIPE memory initialization status register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, PARITY
4Ch	INT_SEC_STS	Interrupt security status register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, PARITY
50h	INT_SEC_CLR	Interrupt security clear register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK
60h	RTINT_SP_L	RTINT stack limit register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK
64h	RTINT_SP_H	RTINT stack limit register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, PARITY
68h	RTISP_STS	RTINT Stack pointer violation status register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK
6Ch	INTSP	INT stack pointer	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, PARITY
80h	LOCK	PIPE lock register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, KEY:KEY=0x5a5a, PARITY
84h	COMMIT	PIPE commit register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, KEY:KEY=0x5a5a, PARITY

Table 6-7. PIPE_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
90h	TASK_CTRL	Task control register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, KEY:KEY=0xcafe, PARITY
94h	BOOT_LINK_CTRL	Boot Link control to lock out BOOT_LINK	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, KEY:KEY=0xface, PARITY
98h	INT_VECT_MAPPING	Interrupt vector mapping for LFU, FOTA, and INT vector swapping support	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, KEY:KEY=0xcafe, PARITY
A0h	MMR_CLR	PIPE MMR clear register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK
A4h	ALL_FLAG_CLR	PIPE flag clear register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, KEY:KEY=0xfeed, PARITY
B0h	REG_PARITY_DIAG_DATA	Register parity Diagnostic data	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK
B8h	REG_PARITY_DIAG_PARITY	Register parity Diagnostic Parity	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK
C0h	REG_PARITY_DIAG_ASSERT	Register parity Assert diagnostic	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, KEY
C8h	REG_PARITY_CHECK	Enabling the Parity check	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, KEY, PARITY
CCh	REG_PARITY_READ	Enabling the Parity read	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, KEY, PARITY
1000h + formula	INT_CTL_L_y	Interrupt low flag and status control register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, , PARITY

Table 6-7. PIPE_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
2000h + formula	INT_CTL_H_y	Interrupt high flag and status control register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, , PARITY
3000h + formula	INT_CONFIG_y	Interrupt configuration register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, LOCK: LOCK.CONFIG_LOCK , PARITY
4000h + formula	INT_LINK_OWNER_y	Interrupt link ownership config register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, LOCK: LOCK.LINK_LOCK
5000h + formula	INT_VECT_ADDR_y	Interrupt vector address	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, LOCK: LOCK.VECT_LOCK
6000h + formula	INT_LINK_OWNER_LFU_y	Interrupt link ownership config register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, LOCK: LOCK.LINK_LOCK
7000h + formula	INT_VECT_ADDR_LFU_y	Interrupt vector address	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK, LOCK: LOCK.VECT_LOCK
8000h	SELFTEST_DIAG_DATA0	Diagnostics data register 0	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK
8004h	SELFTEST_DIAG_DATA1	Diagnostics data register 1	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK
8020h	SELFTEST_DIAG_ECC	Diagnostics ECC	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK
8028h	SELFTEST_DIAG_CONTROL	Diagnostic test enable.	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK
802Ch	SELFTEST_DIAG_STATUS	Diagnostic status register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK

Table 6-7. PIPE_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
8030h	SELFTEST_DIAG_STATUS_CLR	Diagnostic status clear register	SROOT_LINK_RW, BOOT_LINK_RW: BOOT_LINK_CTRL. BOOT_LINK_LOCK

Complex bit access types are encoded to fit into small table cells. Table 6-8 shows the codes that are used for access types in this section.

Table 6-8. PIPE_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

6.7.2.1 RTINT_THRESHOLD Register (Offset = 0h) [Reset = 00000000h]

RTINT_THRESHOLD is shown in Figure 6-5 and described in Table 6-9.

Return to the Summary Table.

Interrupt threshold register

Figure 6-5. RTINT_THRESHOLD Register

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

Table 6-9. RTINT_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	INT_RTINT_THRESHOLD	R/W	0h	Interrupt threshold value to assign an incoming interrupt as a RTINT, INT, or Supervisor INT interrupt. If INT_CONFIGx.PRI_LEVEL < INT_RTINT_THRESHOLD, the incoming interrupt is a RTINT. If INT_CONFIGx.PRI_LEVEL >= INT_RTINT_THRESHOLD, the incoming interrupt is an INT. If INT_CONFIGx.PRI_LEVEL == INT_RTINT_THRESHOLD and TASK_CTRL.SUP_IGN_INTE_EN is set, the incoming interrupt is a Supervisor INT. Reset type: SYSRSn

6.7.2.2 INT_GRP_MASK Register (Offset = 4h) [Reset = 000000FFh]

INT_GRP_MASK is shown in Figure 6-6 and described in Table 6-10.

Return to the Summary Table.

Interrupt group mask register

Figure 6-6. INT_GRP_MASK Register

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

Table 6-10. INT_GRP_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	INT_GRP_MASK	R/W	FFh	Interrupt group mask bits. 0xFF: All interrupts can nest in one another. All interrupt levels are in individual groups, so 256 configurable priorities are available with 1 interrupt level in each. 0xFE:128 priority groups with 2 interrupt levels in each, hence the number of priorities reduce to 128 for preemption. 0xFC: 64 priority groups with 4 interrupt levels in each, hence the number of priorities reduce to 64 for preemption. 0xF8: 32 priority groups with 8 interrupt levels in each, hence the number of priorities reduce to 32 for preemption. 0xF0: 16 priority groups with 16 interrupt levels in each, hence the number of priorities reduce to 16 for preemption. 0xE0: 8 priority groups with 32 interrupt levels in each, hence the number of priorities reduce to 8 for preemption. 0xC0: 4 priority groups with 64 interrupt levels in each, hence the number of priorities reduce to 4 for preemption. 0x80: 2 priority groups with 128 interrupt levels in each, hence the number of priorities reduce to 2 for preemption. 0x00: No interrupts can nest in one another. All interrupt levels are in one group, so configurable priorities are used for the highest priority interrupt. There is no nesting of interrupt levels regardless of INT or RTINT status. All other values: If any other value is used, the mask value defaults to 0xFF which correlates to no group mask. Reset type: SYSRSn

6.7.2.3 GLOBAL_EN Register (Offset = 8h) [Reset = 00000000h]

GLOBAL_EN is shown in Figure 6-7 and described in Table 6-11.

Return to the Summary Table.

Global enable for INT and RTINT

Figure 6-7. GLOBAL_EN Register

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

Table 6-11. GLOBAL_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write Key. 0xFACE Writes to all bits in this register are enabled. Other Values: Writes to any bits in this register are ignored, including separate 16-bit writes. NOTE: The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	ENABLE	R/W	0h	Global enable bit to disable forwarding of INT and RTINT interrupts to CPU. Write of '11' will enable INT & RTINT. Note: 1. Does not impact arbitration. 2. Enables/disables should be within the ATOMIC block of CPU to have predictable behavior of PIPE and CPU interrupt response. Reset type: SYSRSn

6.7.2.4 REVISION Register (Offset = Ch) [Reset = 00000000h]

REVISION is shown in Figure 6-8 and described in Table 6-12.

Return to the Summary Table.

Reserved

Figure 6-8. REVISION Register

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

Table 6-12. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	RESERVED	R	0h	Reserved
15-8	MAJREV	R	0h	This hardcoded field defines the major revision of the IP. Reset type: SYSRSn
7-0	MINREV	R	0h	This hardcoded field defines the minor revision of the IP. Reset type: SYSRSn

6.7.2.5 CPU_INT_STS Register (Offset = 20h) [Reset = 00000000h]

CPU_INT_STS is shown in Figure 6-9 and described in Table 6-13.

Return to the Summary Table.

CPU interrupt status

Figure 6-9. CPU_INT_STS Register

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

Table 6-13. CPU_INT_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	LAST_RTINT_PRIOLVL	R	0h	Priority level of the last RTINT interrupt in service captured when DSTS.INTS='10'. Reset type: SYSRSn
15-8	LAST_INT_PRIOLVL	R	0h	Priority level of the last INT interrupt in service captured when DSTS.INTS='01'. Reset type: SYSRSn
7-2	RESERVED	R	0h	Reserved
1-0	LAST_INTSTS	R	0h	Captures the DSTS.INTS values when INTS = '01' or '10' to track the last serviced interrupt type as either INT or RTINT. NOTE: For current status of the CPU, check DSTS.INTS Reset type: SYSRSn

6.7.2.6 RST_VECT Register (Offset = 24h) [Reset = X0000000h]

RST_VECT is shown in Figure 6-10 and described in Table 6-14.

Return to the Summary Table.

Reset vector register

Figure 6-10. RST_VECT Register

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

Table 6-14. RST_VECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VECT	R	Xh	Reset vector location where the CPU branches on reset de-assertion. For CPU1, this field is set as 0x0 as input from BOOTROM. For CPU2/3, this field is input from the SSU. Reset type: SYSRSn

6.7.2.7 RST_LINK_OWNER Register (Offset = 28h) [Reset = 0000000Xh]

RST_LINK_OWNER is shown in Figure 6-11 and described in Table 6-15.

Return to the Summary Table.

Reset link owner register

Figure 6-11. RST_LINK_OWNER Register

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

Table 6-15. RST_LINK_OWNER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	LINK_OWNER	R	Xh	Reset vector link owner. For CPU1, this field is set as 0x0 as input from BOOTROM. For CPU2/3, this field is input from the SSU. Reset type: SYSRSn

6.7.2.8 NMI_STS Register (Offset = 2Ch) [Reset = 00000000h]

NMI_STS is shown in Figure 6-12 and described in Table 6-16.

Return to the Summary Table.

Non Maskable Interrupt status register

Figure 6-12. NMI_STS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						NMI_CLR	NMI_FLAG
R-0h						R-0/W1C-0h	R-0h

Table 6-16. NMI_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	NMI_CLR	R-0/W1C	0h	Clear active status of NMI_FLAG. Reset type: SYSRSn
0	NMI_FLAG	R	0h	Indicates when the NMI interrupt line is active to the CPU. This field is automatically cleared when the CPU enters the NMI ISR (ACK from the CPU). NMI behaves similar to a pulse interrupt. Reset type: SYSRSn

6.7.2.9 NMI_VECT Register (Offset = 30h) [Reset = X0000000h]

NMI_VECT is shown in Figure 6-13 and described in Table 6-17.

Return to the Summary Table.

Non Maskable Interrupt vector register

Figure 6-13. NMI_VECT Register

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

Table 6-17. NMI_VECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VECT	R/W	Xh	NMI vector location. This field can be written by SW. For CPU1, this field is set as 0x40 as input from BOOTROM. For CPU2/3, this field is input from the SSU. Reset type: SYSRSn

6.7.2.10 NMI_LINK_OWNER Register (Offset = 34h) [Reset = 0000000Xh]

NMI_LINK_OWNER is shown in Figure 6-14 and described in Table 6-18.

Return to the Summary Table.

Non Maskable Interrupt link owner register

Figure 6-14. NMI_LINK_OWNER Register

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

Table 6-18. NMI_LINK_OWNER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	LINK_OWNER	R/W	Xh	NMI vector link owner. For CPU1, this field is set as 0x0 as input from BOOTROM. For CPU2/3, this field is input from the SSU. Reset type: SYSRSn

6.7.2.11 MEM_ECC_DIAG Register (Offset = 40h) [Reset = 00000000h]

MEM_ECC_DIAG is shown in Figure 6-15 and described in Table 6-19.

Return to the Summary Table.

ECC diagnostics register

Figure 6-15. MEM_ECC_DIAG Register

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

Table 6-19. MEM_ECC_DIAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write Key. 0x5A5A: Writes to all bits in this register are enabled. Other Values: Writes to any bits in this register are ignored, including separate 16-bit writes. NOTE: The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
15-2	RESERVED	R	0h	Reserved
1	MEM_SIC_DIAG_EN	R/W	0h	0x0: Memory safe interconnect diagnostics disabled. Access to Memory will be normal. 0x1: Memory safe interconnect diagnostics enable. When enabled user can write to vector memory without altering ECC fields to insert single bit/ double bit errors. This enables to insert errors in fields. Reset type: SYSRSn
0	ECC_VIEW	R/W	0h	0x0: Read on INT_{#}_VECT_ADDR address will return 32 bit vector address 0x1: Read on INT_{#}_VECT_ADDR will reflect ECC value on [6:0] and remaining bits will be Zero Reset type: SYSRSn

6.7.2.12 MEM_INIT Register (Offset = 44h) [Reset = 00000000h]

MEM_INIT is shown in Figure 6-16 and described in Table 6-20.

Return to the Summary Table.

PIPE vector memory initialization register

Figure 6-16. MEM_INIT Register

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

Table 6-20. MEM_INIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write Key. 0x5A5A: Writes to all bits in this register are enabled. Other Values: Writes to any bits in this register are ignored, including separate 16-bit writes. NOTE: The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	INIT	R-0/W	0h	Memory initialization start. 2'b11: Triggers vector memory initialization which is done by BOOTROM. NOTE: By default memory initialization is not asserted on reset de-assertion. SW needs to initiate by writing 2'b11 to this field. Reset type: SYSRSn

6.7.2.13 MEM_INIT_STS Register (Offset = 48h) [Reset = 00000000h]

MEM_INIT_STS is shown in Figure 6-17 and described in Table 6-21.

Return to the Summary Table.

PIPE memory initialization status register

Figure 6-17. MEM_INIT_STS Register

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

Table 6-21. MEM_INIT_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	INIT_STS	R	0h	Memory initialization. 2'b00 : Reset value 2'b01 : Memory initialization is ongoing. 2'b10 : Memory initialization completed. 2'b11 : Reserved Reset type: SYSRSn

6.7.2.14 INT_SEC_STS Register (Offset = 4Ch) [Reset = 00000000h]

INT_SEC_STS is shown in Figure 6-18 and described in Table 6-22.

Return to the Summary Table.

Interrupt security status register

Figure 6-18. INT_SEC_STS Register

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

Table 6-22. INT_SEC_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_FAIL_FLAG	R	0h	VBUSP security fail status flag outputs an error when access permissions to PIPE memory mapped registers and vector tables are violated. Check the details of security privileges for different register types. Reset type: SYSRSn

6.7.2.15 INT_SEC_CLR Register (Offset = 50h) [Reset = 00000000h]

INT_SEC_CLR is shown in Figure 6-19 and described in Table 6-23.

Return to the Summary Table.

Interrupt security clear register

Figure 6-19. INT_SEC_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							SEC_FAIL_FLAG_CLR
R-0h							R-0/W1C-0h

Table 6-23. INT_SEC_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_FAIL_FLAG_CLR	R-0/W1C	0h	Clear bit for SEC_FAIL_FLAG status bit. Reset type: SYSRSn

6.7.2.16 RTINT_SP_L Register (Offset = 60h) [Reset = 00000F00h]

RTINT_SP_L is shown in Figure 6-20 and described in Table 6-24.

Return to the Summary Table.

RTINT stack limit register

Figure 6-20. RTINT_SP_L Register

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

Table 6-24. RTINT_SP_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	Reserved
12-8	MAXRTISP	R	Fh	Maximum Real Time Interrupt Stack Pointer comparison point to stop RTINT nesting and issue an NMI. Reset type: SYSRSn
7-5	RESERVED	R	0h	Reserved
4-0	RTISP	R	0h	Real Time Interrupt Stack Pointer. Reset type: SYSRSn

6.7.2.17 RTINT_SP_H Register (Offset = 64h) [Reset = 0000000Fh]

RTINT_SP_H is shown in Figure 6-21 and described in Table 6-25.

Return to the Summary Table.

RTINT stack limit register

Figure 6-21. RTINT_SP_H Register

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

Table 6-25. RTINT_SP_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	WARNRTISP_PRIO_LEVEL	R/W	0h	RTINT Stack Pointer warning priority level to be used as a threshold for forwarding RTINT upon RTINT stack pointer reaches warning level. Only interrupts of higher priority than this level are sent to CPU. Reset type: SYSRSn
7-5	RESERVED	R	0h	Reserved
4-0	WARNRTISP	R/W	Fh	Warning Real Time Interrupt Stack Pointer comparison point to slow down RTINT nesting and stack growth as a warning for application. NOTE: With the default value of 0xF, PIPE does not generate an interrupt as a warning and will directly generate an interrupt with MAXRTISP. Reset type: SYSRSn

6.7.2.18 RTISP_STS Register (Offset = 68h) [Reset = 00000000h]

RTISP_STS is shown in Figure 6-22 and described in Table 6-26.

Return to the Summary Table.

RTINT Stack pointer violation status register

Figure 6-22. RTISP_STS Register

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

Table 6-26. RTISP_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	MAXRTISP_STS	R	0h	RTINT Stack Pointer max breach status. Reset type: SYSRSn
0	WARNRTISP_STS	R	0h	RTINT Stack Pointer warning status. Reset type: SYSRSn

6.7.2.19 INTSP Register (Offset = 6Ch) [Reset = 00000000h]

INTSP is shown in Figure 6-23 and described in Table 6-27.

Return to the Summary Table.

INT stack pointer

Figure 6-23. INTSP Register

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

Table 6-27. INTSP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	INTSP	R/W	0h	This register provides the stack pointer for INT. CPU will only acknowledge the INT when its current execution stack matched with INTSP. Reset type: SYSRSn

6.7.2.20 LOCK Register (Offset = 80h) [Reset = 00000000h]

LOCK is shown in Figure 6-24 and described in Table 6-28.

Return to the Summary Table.

PIPE lock register

Figure 6-24. LOCK Register

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

Table 6-28. LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write Key. 0x5A5A: Writes to all bits in this register are enabled. Other Values: Writes to any bits in this register are ignored, including separate 16-bit writes. NOTE: The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3	VECT_LOCK	R/W	0h	Lock bit for all INT_{#}_VECT_ADDR registers. 0x0: Writes are allowed to all INT_{#}_VECT_ADDR registers with appropriate permissions. 0x1: Writes are not allowed to any INT_{#}_VECT_ADDR. Reset type: SYSRSn
2	GLOBAL_LOCK	R/W	0h	Lock bit for following registers: 1. RTINT_THRESHOLD 2. INT_GRP_MASK 3. GLOBAL_EN 4. NMI_VECT 5. NMI_LINK 0x0: Writes are allowed to above registers. 0x1: Writes are not allowed to above registers. Reset type: SYSRSn
1	CONFIG_LOCK	R/W	0h	Lock bit for all INT_{#}_CONFIG registers 0x0: Writes are allowed to all INT_{#}_CONFIG registers with appropriate permissions. 0x1: Writes are not allowed to any INT_{#}_CONFIG. Reset type: SYSRSn

Table 6-28. LOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	LINK_LOCK	R/W	0h	Lock bit for all INT_{#}_LINK_OWNER registers. 0x0: Writes are allowed to all INT_{#}_LINK_OWNER registers with appropriate permissions. 0x1: Writes are not allowed to any INT_{#}_LINK_OWNER. Reset type: SYSRSn

6.7.2.21 COMMIT Register (Offset = 84h) [Reset = 00000000h]

COMMIT is shown in Figure 6-25 and described in Table 6-29.

Return to the Summary Table.

PIPE commit register

Figure 6-25. COMMIT Register

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

Table 6-29. COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write Key. 0x5A5A: Writes to all bits in this register are enabled. Other Values: Writes to any bits in this register are ignored, including separate 16-bit writes. NOTE: The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3	VECT_COMMIT	R/WSonce	0h	Commit bit for LOCK.VECT_LOCK. 0x0: Writes to LOCK.VECT_LOCK register bit is allowed. 0x1: Writes to LOCK.VECT_LOCK register bit is not allowed. Reset type: SYSRSn
2	GLOBAL_COMMIT	R/WSonce	0h	Commit bit for LOCK.GLOBAL_LOCK. 0x0: Writes to LOCK.GLOBAL_LOCK register bit is allowed. 0x1: Writes to LOCK.GLOBAL_LOCK register bit is not allowed. Reset type: SYSRSn
1	CONFIG_COMMIT	R/WSonce	0h	Commit bit for LOCK.CONFIG_LOCK. 0x0: Writes to LOCK.CONFIG_LOCK register bit is allowed. 0x1: Writes to LOCK.CONFIG_LOCK register bit is not allowed. Reset type: SYSRSn
0	LINK_COMMIT	R/WSonce	0h	Commit bit for LOCK.LINK_LOCK register bit. 0x0: Writes to LOCK.LINK_LOCK register bit is allowed. 0x1: Writes to LOCK.LINK_LOCK register bit is not allowed. Reset type: SYSRSn

6.7.2.22 TASK_CTRL Register (Offset = 90h) [Reset = 00000000h]

TASK_CTRL is shown in Figure 6-26 and described in Table 6-30.

Return to the Summary Table.

Task control register

Figure 6-26. TASK_CTRL Register

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

Table 6-30. TASK_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write Key. 0xCAFE: Writes to all bits in this register are enabled. Other Values: Writes to any bits in this register are ignored, including separate 16-bit writes. NOTE: The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
15-9	RESERVED	R	0h	Reserved
8	SUP_IGN_INTE_EN	R/W	0h	This field enables the highest priority INT interrupt to be used without regard to DSTS.INTE enable within the CPU. Reset type: SYSRSn
7-2	RESERVED	R	0h	Reserved
1-0	ACTIVE_CONTEXT_ID	R/W	0h	This field acts as a mask for enabling the interrupt in arbitration participation or not. Each interrupt line has CONTEXT_ID in its configuration register. Interrupts for which this field matches the corresponding CONTEXT_ID participate in interrupt arbitration with this decoding: '00' - Context 0 - Default context where all interrupts are in this context. '01' - Context 1 '10' - Context 2 '11' - Reserved Reset type: SYSRSn

6.7.2.23 BOOT_LINK_CTRL Register (Offset = 94h) [Reset = 00000000h]

BOOT_LINK_CTRL is shown in Figure 6-27 and described in Table 6-31.

Return to the Summary Table.

Boot Link control to lock out BOOT_LINK

Figure 6-27. BOOT_LINK_CTRL Register

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

Table 6-31. BOOT_LINK_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write Key. 0xFACE: Writes to all bits in this register are enabled. Other Values: Writes to any bits in this register are ignored, including separate 16-bit writes. NOTE: The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
15-1	RESERVED	R	0h	Reserved
0	BOOT_LINK_LOCK	R/WSonce	0h	This register when set, locks any further write access from BOOT LINK to the PIPE configuration registers. Once the bit is set, it has no effect of any write or any other change till the next reset. Reset type: SYSRSn

6.7.2.24 INT_VECT_MAPPING Register (Offset = 98h) [Reset = 00000000h]

INT_VECT_MAPPING is shown in Figure 6-28 and described in Table 6-32.

Return to the Summary Table.

Interrupt vector mapping for LFU, FOTA, and INT vector swapping support

Figure 6-28. INT_VECT_MAPPING Register

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

Table 6-32. INT_VECT_MAPPING Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write Key. 0xCAFE: Writes to all bits in this register are enabled. Other Values: Writes to any bits in this register are ignored, including separate 16-bit writes. NOTE: The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	LFU_INT_VECT_MAPPING	R/W	0h	This field controls the selection between PIPE_VECT_MEM and PIPE_VECT_LFU_MEM to support live firmware update (LFU) operation to get vector values. This value doesn't have any impact on direct CPU access to PIPE vector addresses and registers through VBUSP interface. 2'b11: Configured as PIPE_VECT_LFU_MEM to support LFU operation Other Values: Configured as PIPE_VECT_MEM by default Reset type: SYSRSn

6.7.2.25 MMR_CLR Register (Offset = A0h) [Reset = 00000000h]

MMR_CLR is shown in Figure 6-29 and described in Table 6-33.

Return to the Summary Table.

PIPE MMR clear register

Figure 6-29. MMR_CLR Register

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

Table 6-33. MMR_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	MMR_CLR	R-0/W	0h	This field resets all the registers when configured: 0x3 : Clears registers. Other value : No impact. List of registers which are impacted by this field are: 1. RTINT_THRESHOLD 2. INT_GRP_MASK 3. LOCK 4. COMMIT 5. TASK_CTRL 6. ALL_FLG_CLR 7. MEM_ECC_DIAG 8. GLOBAL_EN 9. INT_CTL_L 10. INT_CFG 11. INT_VECT_MAPPING 12. REG_PARITY_DIAG_DATA 13. REG_PARITY_DIAG_PARITY_DATA 14. REG_PARITY_DIAG_ASSERT 15. REG_PARITY_CHECK 16. REG_PARITY_READ Reset type: SYSRSn

6.7.2.26 ALL_FLAG_CLR Register (Offset = A4h) [Reset = 00000000h]

ALL_FLAG_CLR is shown in Figure 6-30 and described in Table 6-34.

Return to the Summary Table.

PIPE flag clear register

Figure 6-30. ALL_FLAG_CLR Register

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

Table 6-34. ALL_FLAG_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write Key. 0xFEED: Writes to all bits in this register are enabled. Other Values: Writes to any bits in this register are ignored, including separate 16-bit writes. NOTE: The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	ALL_FLAG_CLR	R/W	0h	This register clears the flags bit fields for all the interrupts in the INT_{#}_CTL_REG_L register. 2'b11: Enable, clears flags. Other values: No impact. Note: This is a R/W type register. User needs to explicitly write back '0' to start capturing interrupts. Reset type: SYSRSn

6.7.2.27 REG_PARITY_DIAG_DATA Register (Offset = B0h) [Reset = 00000000h]

REG_PARITY_DIAG_DATA is shown in Figure 6-31 and described in Table 6-35.

Return to the Summary Table.

Register parity Diagnostic data

Figure 6-31. REG_PARITY_DIAG_DATA Register

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

Table 6-35. REG_PARITY_DIAG_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DIAG_DATA	R/W	0h	Diagnostics data This register is used to specify the [31:0] bits of the data to perform parity diagnostics. Reset type: SYSRSn

6.7.2.28 REG_PARITY_DIAG_PARITY Register (Offset = B8h) [Reset = 00000000h]

REG_PARITY_DIAG_PARITY is shown in Figure 6-32 and described in Table 6-36.

Return to the Summary Table.

Register parity Diagnostic Parity

Figure 6-32. REG_PARITY_DIAG_PARITY Register

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

Table 6-36. REG_PARITY_DIAG_PARITY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	DIAG_PARITY_DATA	R/W	0h	Diagnostics parity data This register is used to specify the [31:0] bits of the parity data to perform parity diagnostics. Reset type: SYSRSn

6.7.2.29 REG_PARITY_DIAG_ASSERT Register (Offset = C0h) [Reset = 00000000h]

REG_PARITY_DIAG_ASSERT is shown in Figure 6-33 and described in Table 6-37.

Return to the Summary Table.

Register parity Assert diagnostic

Figure 6-33. REG_PARITY_DIAG_ASSERT Register

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
KEY							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							DIAG_ASSERT
R-0h							R-0/W1S-0h

Table 6-37. REG_PARITY_DIAG_ASSERT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write Key. 0x5A5A: Writes to all bits in this register are enabled. Other Values: Writes to any bits in this register are ignored, including separate 16-bit writes. NOTE: The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
15-1	RESERVED	R	0h	Reserved
0	DIAG_ASSERT	R-0/W1S	0h	Diagnostics assert This register is used to assert parity diagnostics. Reset type: SYSRSn

6.7.2.30 REG_PARITY_CHECK Register (Offset = C8h) [Reset = 00000000h]

REG_PARITY_CHECK is shown in Figure 6-34 and described in Table 6-38.

Return to the Summary Table.

Enabling the Parity check

Figure 6-34. REG_PARITY_CHECK Register

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

Table 6-38. REG_PARITY_CHECK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write Key. 0x5A5A: Writes to all bits in this register are enabled. Other Values: Writes to any bits in this register are ignored, including separate 16-bit writes. NOTE: The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3-0	MODE	R/W	0h	0000: Disabled 1010: One Shot mode All other values: Continuous mode Note: In One Shot mode, HW will clear this register ('0000') once it completed the parity check of all the registers (counter reaches it's maximum value). SW needs to write into this field again for next check. Reset type: SYSRSn

6.7.2.31 REG_PARITY_READ Register (Offset = CCh) [Reset = 00000000h]

REG_PARITY_READ is shown in Figure 6-35 and described in Table 6-39.

Return to the Summary Table.

Enabling the Parity read

Figure 6-35. REG_PARITY_READ Register

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

Table 6-39. REG_PARITY_READ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write Key. 0x5A5A: Writes to all bits in this register are enabled. Other Values: Writes to any bits in this register are ignored, including separate 16-bit writes. NOTE: The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3-0	PARITY_READ_EN	R/W	0h	1010: Parity read is enabled All other values : Parity read is disabled. Note: When the parity read is enabled, actual registers are not accessible for read in the memory map. Instead, the parity values are accessible. Reset type: SYSRSn

6.7.2.32 INT_CTL_L_y Register (Offset = 1000h + formula) [Reset = 00000000h]

INT_CTL_L_y is shown in Figure 6-36 and described in Table 6-40.

Return to the Summary Table.

Interrupt low flag and status control register

Offset = 1000h + (y * 4h); where y = 0h to FFh

Figure 6-36. INT_CTL_L_y Register

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

Table 6-40. INT_CTL_L_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	OVERFLOW_FLAG	R	0h	Interrupt overflow flag status. This will be set when there is an new interrupt request from peripheral before an ACK from the CPU. Note: This flag is only set if the FLAG is already set and another interrupt occurs and applicable for pulse interrupts Reset type: SYSRSn
1	FLAG	R	0h	Interrupt active register. This field will be set on either an active interrupt from peripherals or SW write to FLAG_FRC bit. Reset type: SYSRSn
0	EN	R/W	0h	Interrupt enable bit. Interrupt line will participate in priority arbitration only when this bit is set. Reset type: SYSRSn

6.7.2.33 INT_CTL_H_y Register (Offset = 2000h + formula) [Reset = 00000000h]

INT_CTL_H_y is shown in Figure 6-37 and described in Table 6-41.

Return to the Summary Table.

Interrupt high flag and status control register

Offset = 2000h + (y * 4h); where y = 0h to FFh

Figure 6-37. INT_CTL_H_y Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					OVERFLOW_FLAG_CLR	FLAG_CLR	FLAG_FRC
R-0h					R-0/W1C-0h	R-0/W1C-0h	R-0/W1S-0h

Table 6-41. INT_CTL_H_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	OVERFLOW_FLAG_CLR	R-0/W1C	0h	Clear bit for overflow flag status bit. Reset type: SYSRSn
1	FLAG_CLR	R-0/W1C	0h	Clear Active status of interrupt. SW Clear will have priority over SW force. Reset type: SYSRSn
0	FLAG_FRC	R-0/W1S	0h	Force set active flag of interrupt. SW Clear will have priority over SW force. Reset type: SYSRSn

6.7.2.34 INT_CONFIG_y Register (Offset = 3000h + formula) [Reset = 000000FFh]

INT_CONFIG_y is shown in Figure 6-38 and described in Table 6-42.

Return to the Summary Table.

Interrupt configuration register

Offset = 3000h + (y * 4h); where y = 0h to FFh

Figure 6-38. INT_CONFIG_y Register

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

Table 6-42. INT_CONFIG_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17-16	CONTEXT_ID	R/W	0h	Interrupt Context-ID. When this field associated with an individual interrupt matches the ACTIVE_CONTEXT_ID field at the PIPE module level, the interrupt is raised in that specific context. '00': Context-0 - Used for default context and all interrupts. '01': Context-1 '10': Context-2 '11': Used for interrupts which are context-agnostic and those participate in arbitration regardless of active context-id. Reset type: SYSRSn
15-8	RESERVED	R	0h	Reserved
7-0	PRI_LEVEL	R/W	FFh	Priority level for INT_{#}_. Reset type: SYSRSn

6.7.2.35 INT_LINK_OWNER_y Register (Offset = 4000h + formula) [Reset = 00000000h]

INT_LINK_OWNER_y is shown in Figure 6-39 and described in Table 6-43.

Return to the Summary Table.

Interrupt link ownership config register

Offset = 4000h + (y * 4h); where y = 0h to FFh

Figure 6-39. INT_LINK_OWNER_y Register

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

Table 6-43. INT_LINK_OWNER_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	Reserved
8	API_LINK_EN	R/W	0h	Access protection inheritance link for INT_{#}_ enable. When API_LINK_EN = 0, check only the OWNER_LINK for writes to INT_{#}_CTL_REG. The API_LINK is not allowed to access PIPE resources associated with interrupt line. When API_LINK_EN = 1, check both the OWNER_LINK & API_LINK for writes to INT_{#}_CTL_REG. The API_LINK is allowed to access PIPE resources associated with interrupt line. Reset type: N/A
7-4	API_LINK	R/W	0h	Access protection inheritance link for INT_{#}_. Reset type: N/A
3-0	OWNER_LINK	R/W	0h	Link owner for INT_{#}_. Reset type: N/A

6.7.2.36 INT_VECT_ADDR_y Register (Offset = 5000h + formula) [Reset = X0000000h]

INT_VECT_ADDR_y is shown in Figure 6-40 and described in Table 6-44.

Return to the Summary Table.

Interrupt vector address

Offset = 5000h + (y * 4h); where y = 0h to FFh

Figure 6-40. INT_VECT_ADDR_y Register

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

Table 6-44. INT_VECT_ADDR_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VECT_ADDR	R/W	Xh	Vector address location of INT_{#}_. For memory initialization, the VECT_ADDR value will be the same as the NMI vector address. For CPU1, this field is set as 0x40 as input from BOOTROM. For CPU2/3, this field is input from the SSU. Reset type: N/A

6.7.2.37 INT_LINK_OWNER_LFU_y Register (Offset = 6000h + formula) [Reset = 00000000h]

INT_LINK_OWNER_LFU_y is shown in Figure 6-41 and described in Table 6-45.

Return to the Summary Table.

Interrupt link ownership config register

Offset = 6000h + (y * 4h); where y = 0h to FFh

Figure 6-41. INT_LINK_OWNER_LFU_y Register

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

Table 6-45. INT_LINK_OWNER_LFU_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	Reserved
8	API_LINK_EN	R/W	0h	Access protection inheritance link for INT_{#}_ enable. When API_LINK_EN = 0, check only the OWNER_LINK for writes to INT_{#}_CTL_REG. The API_LINK is not allowed to access PIPE resources associated with interrupt line. When API_LINK_EN = 1, check both the OWNER_LINK & API_LINK for writes to INT_{#}_CTL_REG. The API_LINK is allowed to access PIPE resources associated with interrupt line. Reset type: N/A
7-4	API_LINK	R/W	0h	Access protection inheritance link for INT_{#}_. Reset type: N/A
3-0	OWNER_LINK	R/W	0h	Link owner for INT_{#}_. Reset type: N/A

6.7.2.38 INT_VECT_ADDR_LFU_y Register (Offset = 7000h + formula) [Reset = X0000000h]

INT_VECT_ADDR_LFU_y is shown in Figure 6-42 and described in Table 6-46.

Return to the Summary Table.

Interrupt vector address

Offset = 7000h + (y * 4h); where y = 0h to FFh

Figure 6-42. INT_VECT_ADDR_LFU_y Register

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

Table 6-46. INT_VECT_ADDR_LFU_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VECT_ADDR	R/W	Xh	Vector address location of INT_{#}_. For memory initialization, the VECT_ADDR value will be the same as the NMI vector address. For CPU1, this field is set as 0x40 as input from BOOTROM. For CPU2/3, this field is input from the SSU. Reset type: N/A

6.7.2.39 SELFTEST_DIAG_DATA0 Register (Offset = 8000h) [Reset = 00000000h]

SELFTEST_DIAG_DATA0 is shown in Figure 6-43 and described in Table 6-47.

Return to the Summary Table.

Diagnostics data register 0

Figure 6-43. SELFTEST_DIAG_DATA0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SELFTEST_DIAG_DATA0																															
R/W-0h																															

Table 6-47. SELFTEST_DIAG_DATA0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SELFTEST_DIAG_DATA0	R/W	0h	Self test Diagnostics data 0. This register is used to specify the [31:0] bits of the data to perform self-test ECC checker diagnostics. Reset type: SYSRSn

6.7.2.40 SELFTEST_DIAG_DATA1 Register (Offset = 8004h) [Reset = 00000000h]

SELFTEST_DIAG_DATA1 is shown in Figure 6-44 and described in Table 6-48.

Return to the Summary Table.

Diagnostics data register 1

Figure 6-44. SELFTEST_DIAG_DATA1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												SELFTEST_DIAG_DATA1																				
R-0h												R/W-0h																			

Table 6-48. SELFTEST_DIAG_DATA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19-0	SELFTEST_DIAG_DATA1	R/W	0h	Self test Diagnostics data 1. This register is used to specify the [19:0] bits of the data to perform self-test ECC checker diagnostics. Reset type: SYSRSn

6.7.2.41 SELFTEST_DIAG_ECC Register (Offset = 8020h) [Reset = 00000000h]

SELFTEST_DIAG_ECC is shown in Figure 6-45 and described in Table 6-49.

Return to the Summary Table.

Diagnostics ECC

Figure 6-45. SELFTEST_DIAG_ECC Register

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

Table 6-49. SELFTEST_DIAG_ECC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6-0	SELFTEST_DIAG_ECC	R/W	0h	Self test Diagnostics ECC. This register is used to specify the ECC to perform self-test ECC checker diagnostics. Reset type: SYSRSn

6.7.2.42 SELFTEST_DIAG_CONTROL Register (Offset = 8028h) [Reset = 34070000h]

SELFTEST_DIAG_CONTROL is shown in Figure 6-46 and described in Table 6-50.

Return to the Summary Table.

Enable diagnostic test

Figure 6-46. SELFTEST_DIAG_CONTROL Register

31	30	29	28	27	26	25	24
DIAG_DATA_WIDTH							
R/W-34h							
23	22	21	20	19	18	17	16
RESERVED				DIAG_ECC_WIDTH			
R-0h				R/W-7h			
15	14	13	12	11	10	9	8
RESERVED				RESERVED			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESERVED				DIAG_TEST_EN			
R-0h				R/W-0h			

Table 6-50. SELFTEST_DIAG_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DIAG_DATA_WIDTH	R/W	34h	PIPE ECC data width needs be configured to 52 (0x34). Reset type: SYSRSn
23-20	RESERVED	R	0h	Reserved
19-16	DIAG_ECC_WIDTH	R/W	7h	PIPE ECC bit width needs to be configured to 7 (0x7). Reset type: SYSRSn
15-12	RESERVED	R	0h	Reserved
11-4	RESERVED	R	0h	Reserved
3-0	DIAG_TEST_EN	R/W	0h	Enable self test mechanism 0011 : Enable self test Any other value will disable self-test This field will be '0000' once test done. User needs to write into this register again for next test. Reset type: SYSRSn

6.7.2.43 SELFTEST_DIAG_STATUS Register (Offset = 802Ch) [Reset = 00000000h]

SELFTEST_DIAG_STATUS is shown in Figure 6-47 and described in Table 6-51.

Return to the Summary Table.

Diagnostic status

Figure 6-47. SELFTEST_DIAG_STATUS Register

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

Table 6-51. SELFTEST_DIAG_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	DIAG_FAIL_BIT_INDEX	R	0h	This field is used to specify the position of the flipped bit when test failed. For 2 bit flips, this field points the bit position of the first bit. The second bit will be always adjacent to the first bit. This field will clear when next test configured. Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6-5	DIAG_FAIL_CHECK_TYPE	R	0h	00 : Positive check 01 : Flips one bit 10 : Flips two bit 11 : Reserved Reset type: SYSRSn
4	DIAG_FAIL_UC_ERROR	R	0h	This field is used to specify the diagnostic uncorrectable error when Test failed. Reset type: SYSRSn
3	DIAG_FAIL_C_ERROR	R	0h	This field is used to specify the diagnostic correctable error when Test failed. Reset type: SYSRSn
2	DIAG_TEST_FAIL	R	0h	1 : Test failed (Unexpected error events(C_ERROR/UC_ERROR) occurred during self test) 0 : Test passed Reset type: SYSRSn
1	DIAG_TEST_DONE	R	0h	Completed self test. Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

6.7.2.44 SELFTEST_DIAG_STATUS_CLR Register (Offset = 8030h) [Reset = 00000000h]

SELFTEST_DIAG_STATUS_CLR is shown in Figure 6-48 and described in Table 6-52.

Return to the Summary Table.

Diagnostic status clear

Figure 6-48. SELFTEST_DIAG_STATUS_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					DIAG_TEST_FAIL	DIAG_TEST_DONE	RESERVED
R-0h					R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 6-52. SELFTEST_DIAG_STATUS_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	DIAG_TEST_FAIL	R-0/W1S	0h	Clear Test failed status flags 0: Writing a 0 has no effect. 1: Writing a 1 will clear the following bits: 1. SELFTEST_DIAG_STATUS[DIAG_TEST_FAIL], 2. SELFTEST_DIAG_STATUS[DIAG_FAIL_C_ERROR], 3. SELFTEST_DIAG_STATUS[DIAG_FAIL_UC_ERROR], 4. SELFTEST_DIAG_STATUS[DIAG_FAIL_CHECK_TYPE], 5. SELFTEST_DIAG_STATUS[DIAG_FAIL_BIT_INDEX]. Reset type: SYSRSn
1	DIAG_TEST_DONE	R-0/W1S	0h	Clear self test done status flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the SELFTEST_DIAG_STATUS[DIAG_TEST_DONE] bit. Reset type: SYSRSn
0	RESERVED	R-0/W1S	0h	Reserved

Chapter 7

Error Signaling Module (ESM_C29)

This chapter describes the features and operation of the error signaling module (ESM_C29) module.

7.1 Introduction

The Error Signaling Module (ESM) provides systematic consolidation of responses to error events throughout the device into one location. The Module can signal programmable priority interrupts to the processor to deal with an event and/or manipulate an I/O error pin to signal an external hardware that an error has occurred. Therefore, an external controller is able to reset the device or keep the system in a safe, known state.

7.1.1 Features

The following points highlights the main features of the Error Signaling Module Subsystem :

- ESM Subsystem consists of three ESM CPU Instances (one for each CPU), a single System ESM (SYS ESM) for Error Pin Output and Monitoring, and an additional Safety Aggregator (ESM Register Parity Error Aggregator) Instance for Register Parity Error Aggregation for all ESM instances (ESM CPU and System ESM).
- Supports up to 256 error event inputs: Error events from Error Aggregator and System level fault event sources are mapped to ESM subsystem and applied to all ESM instances (Refer to Section 7.3.1 for more details)
  - Divided in groups of 32 error events (Total 8 Groups)
- Selectable Low and High Priority Interrupt prioritization of each error event for each ESM CPU instance
- Critical Priority and High Priority Watchdog Interrupt outputs for each ESM CPU instance
  - Pulse type output only
- System ESM Instance to signal severe device failure produces an Error Pin Output in addition to set of interrupt outputs
  - Supports of level or PWM modes for error pin output
  - Error Pin Monitoring
- Configurable timebase for error pin signal output
- Error forcing capability
- Parity Detection and Commit/Lock for MMRs

7.1.2 ESM Related Collateral

Foundational Materials

- C29x Academy - Error Handling

7.2 ESM Subsystem

The Error Signaling Module Subsystem (ESM-SS) groups error signaling module (ESM) instances as shown in Figure 7-1. ESMSS supports a number of ESM instances that is triggered from common set of error event inputs. Each ESM instance is used to drive interrupts to individual CPU and resets to individual CPU or System. The subsystem combines the ESM instances and output pulse interrupt from each ESM instances are exported at the subsystem boundary for integration at the device level.

ESM subsystem is comprised of the following instances :

- ESM CPU instances one for each CPU
  - **Input's**: The error inputs listed in Section 7.3.1, common to all ESM Subsystem instances
  - **Output's**:
    - Low Priority Interrupt
    - High Priority Interrupt
    - High Priority WD Event (Event triggered by watchdog timeout on High Priority Interrupt hence also referred to as High Priority Watchdog Interrupt in the later part of the document): Similar functionality as NMIWD on C28x devices
    - Critical Priority Interrupt

- Additional System ESM instance for Error Pin output and monitoring.
  - **Input's**: The error inputs listed in Section 7.3.1, common to all ESM Subsystem instances
  - **Output's**:
    - Low Priority Interrupt
    - Critical Priority Interrupt
    - Error Pin Output
    - Error Pin Monitor Event : Error pin monitoring and error detection output
- Register Parity Error Aggregator Instance (Safety Aggregator)
  - **Input's**:
    - Input from EDC (Error Detection and Correction) Control Interfaces of all ESM Instances (ESM CPU and SYS ESM)
  - **Output**:
    - Parity Error Interrupt : Interrupt generated by parity error detected on ESM register configurations

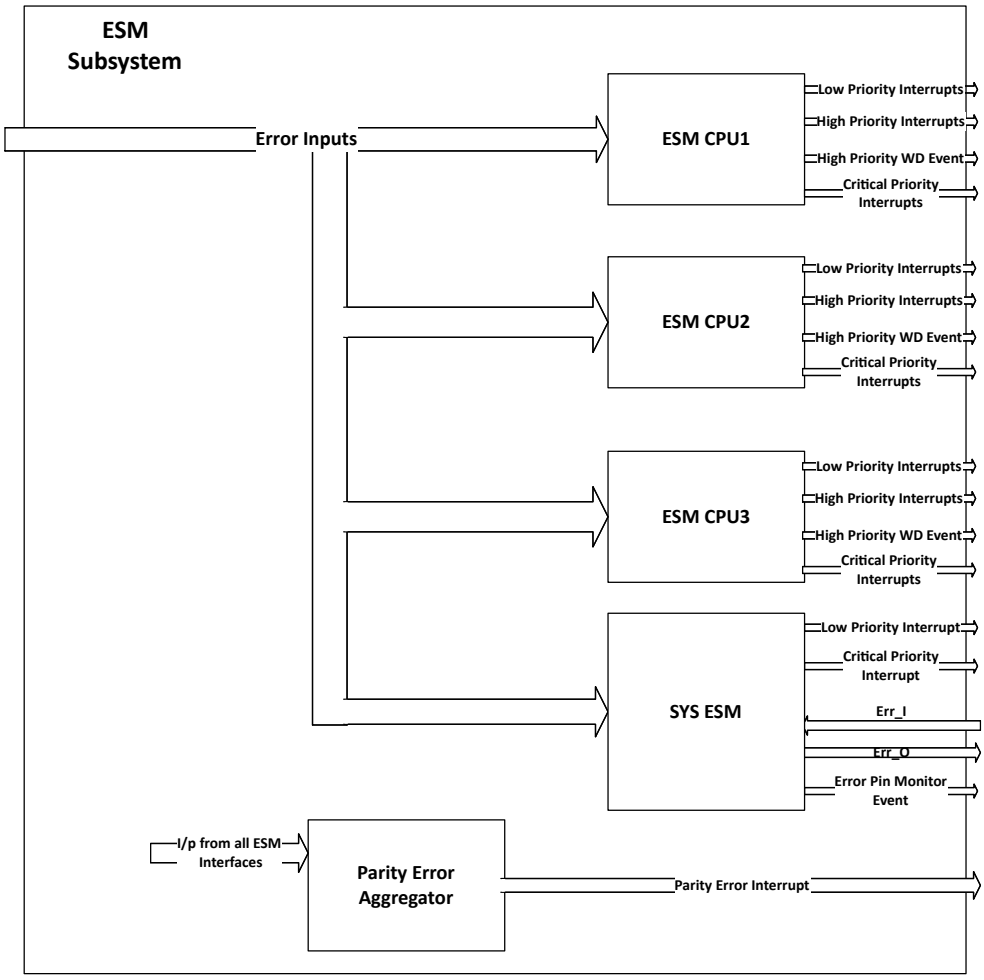


Figure 7-1. ESM-SS Block Diagram

7.2.1 System ESM

Error Pin inputs and outputs are controlled by System ESM instance. The System ESM produces an configurable error pin output (err_o/ERRORSTS) in addition to the set of interrupt outputs. The System ESM generates a critical priority interrupt (ESMRESET) output that causes system reset request (XRSn) by default, if not disabled by the ESMXRSNCTL register. The System ESM additionally has the Error Pin Monitor feature and associated Error Pin Monitor event that is exported at subsystem as a pulse interrupt. Error Pin Monitor event is also applied back to ESM-SS as an error event input so that the ESM can take appropriate action on the mismatch event.

The low-priority interrupt output of System ESM is mapped to the XBARs as an ESMGENEVT signal.

7.2.1.1 Error Pin Monitor Event

The error pin monitor event indicates that the err_i and err_o pins (input and output to the ESM, respectively) have differed for eight or more consecutive cycles. This detects if the error pin connectivity external to the ESM is malfunctioning.

ERR_I is stored, in case previous values of ERR_I are either all 0's or 1's, the previous 8 values of ERR_O is checked for atleast one match. If there is no match between previous 8 values of ERR_I and ERR_O, this indicates that the external connection from ERR_O back to ERR_I is malfunctioning. Error Pin Monitor event/ interrupt is triggered only if all values of ERR_I being equal (all zeroes or ones) and stored ERR_O values being the logical inverse (all zeroes or ones). The above mechanism provides immunity against glitches or pin transitions triggering spurious pin monitor interrupt outputs. PWM low and high counters values need to be greater than 8 cycles when using pwm mode for error pin monitor to detect a malfunctioning ERR_I.

When an error pin monitor interrupt occurs, a processor handling this interrupt logs or alerts the external system associated with error pin monitoring. This interrupt can be enabled/disabled via the Error Pin Monitor Config Register and set or cleared via Error Pin Monitor Interrupt Status/Set Register and Error Pin Monitor Interrupt Status/Clear Register respectively.

A software write of the EOI vector to the EOI Interrupt Register also results in a re-evaluation of the error pin monitor interrupt/event.

7.2.2 Safety Aggregator

Safety Aggregator module is used within the ESM subsystem to aggregate all the error detection and correction (EDC) interfaces from each ESM instance. The ESM instances (including the System ESM instance) each has an EDC Controller to control parity detection logic and injection of errors for testing purposes.

7.2.2.1 EDC Controller Interface Description

Each ESM Instance has a EDC (Error detection and Correction) control interface which works with Safety aggregator for parity detection. Control interface receives command from Safety Aggregator though a serial vbus connection, the error status information can be accessed through the registers described in Section 7.2.2.1.1.

7.2.2.1.1 EDC_REGS Registers

Table 7-1 lists the memory-mapped registers for the EDC_REGS registers. All register offset addresses not listed in Table 7-1 should be considered as reserved locations and the register contents should not be modified.

Table 7-1. EDC_REGS Registers

Offset	Acronym	Register Name	Protection
10h	REVISION	Revision Register	
14h	CONTROL	Control Register	
18h	ERROR1	Error 1 Register	
1Ch	ERROR2	Error 2 Register	
20h	ERRORSTATUS1	Error Status 1 Register	
24h	ERRORSTATUS2	Error Status 2 Register	

Complex bit access types are encoded to fit into small table cells. Table 7-2 shows the codes that are used for access types in this section.

Table 7-2. EDC_REGS Access Type Codes

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

7.2.2.1.1.1 REVISION Register (Offset = 10h) [Reset = 4F404900h]

REVISION is shown in Figure 7-2 and described in Table 7-3.

Return to the Summary Table.

Revision Register

Figure 7-2. REVISION Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCHEME		RSVD		FUNC											
R-1h		R-0h		R-F40h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTL					MAJOR			CUSTOM		MINOR					
R-9h					R-1h			R-0h		R-0h					

Table 7-3. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	Reserved Reset type: SYSRSn
29-28	RSVD	R	0h	Reserved Reset type: SYSRSn
27-16	FUNC	R	F40h	Assigned func id Reset type: SYSRSn
15-11	RTL	R	9h	RTL version of the module Reset type: SYSRSn
10-8	MAJOR	R	1h	Major revision of module Reset type: SYSRSn
7-6	CUSTOM	R	0h	Special version Reset type: SYSRSn
5-0	MINOR	R	0h	Minor revision of module Reset type: SYSRSn

7.2.2.1.1.2 CONTROL Register (Offset = 14h) [Reset = 00000002h]

CONTROL is shown in Figure 7-3 and described in Table 7-4.

Return to the Summary Table.

Control Register

Figure 7-3. CONTROL Register

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

Table 7-4. CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RSVD	R	0h	RESERVED Reset type: SYSRSn
11-8	ECC_PATTERN	R/W	0h	The data pattern to use for injection. 0 = 0s 1 = Fs 2 = As 3 = 5s Reset type: SYSRSn
7-6	RSVD	R	0h	Reset type: SYSRSn
5	FORCE_N_BIT	R/W	0h	Update injection fields after the injection to setup for the next incremental injection. 0 = keep current settings after injection 1 = increment to next bit or group after injection Reset type: SYSRSn
4	FORCE_DE	R/W	0h	Inject a single bit error when set. Will be automatically cleared when injection has completed. Reset type: SYSRSn
3	FORCE_SE	R/W	0h	Inject a double bit error when set. Will be automatically cleared when injection has completed Reset type: SYSRSn
2	RSVD	R	0h	RESERVED Reset type: SYSRSn
1	ECCCHECK	R/W	1h	Enable ECC Checkers 0 = Disabled 1 = Enabled Reset type: SYSRSn
0	RSVD	R	0h	RESERVED Reset type: SYSRSn

7.2.2.1.1.3 ERROR1 Register (Offset = 18h) [Reset = 00000000h]

ERROR1 is shown in Figure 7-4 and described in Table 7-5.

Return to the Summary Table.

Error 1 Register

Figure 7-4. ERROR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							ECCBIT1									ECCGRP															
R-0h							R/W-0h									R/W-0h															

Table 7-5. ERROR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RSVD	R	0h	RESERVED Reset type: SYSRSn
24-16	ECCBIT1	R/W	0h	First bit to inject an error Reset type: SYSRSn
15-0	ECCGRP	R/W	0h	Group of Checker to inject Reset type: SYSRSn

7.2.2.1.1.4 ERROR2 Register (Offset = 1Ch) [Reset = 00000000h]

ERROR2 is shown in Figure 7-5 and described in Table 7-6.

Return to the Summary Table.

Error 2 Register

Figure 7-5. ERROR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							ECCBIT2								
R-0h																							R/W-0h								

Table 7-6. ERROR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RSVD	R	0h	RESERVED Reset type: SYSRSn
8-0	ECCBIT2	R/W	0h	Second bit to inject an error. Only valid if force_de is set. Reset type: SYSRSn

7.2.2.1.1.5 ERRORSTATUS1 Register (Offset = 20h) [Reset = 00000000h]

ERRORSTATUS1 is shown in Figure 7-6 and described in Table 7-7.

Return to the Summary Table.

Error Status 1 Register

Figure 7-6. ERRORSTATUS1 Register

31	30	29	28	27	26	25	24
ERR_GROUP							
R-0h							
23	22	21	20	19	18	17	16
ERR_GROUP							
R-0h							
15	14	13	12	11	10	9	8
INJ_UNCOR_PEND_CLR		INJ_COR_PEND_CLR		UNCOR_PEND_CLR		COR_PEND_CLR	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
INJ_UNCOR_PEND		INJ_COR_PEND		UNCOR_PEND		COR_PEND	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 7-7. ERRORSTATUS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ERR_GROUP	R	0h	Specify Checker that reported Error Reset type: SYSRSn
15-14	INJ_UNCOR_PEND_CLR	R/W	0h	Number of injected uncorrected pending interrupts (same value as ded-pend). Writing decrements inj_ded_pend by that value. Reset type: SYSRSn
13-12	INJ_COR_PEND_CLR	R/W	0h	Number of injected corrected pending interrupts (same value as sec_pend). Writing decrements inj_sec_pend by that value. Reset type: SYSRSn
11-10	UNCOR_PEND_CLR	R/W	0h	Number of uncorrected pending interrupts (same value as ded-pend). Writing decrements ded_pend by that value. Reset type: SYSRSn
9-8	COR_PEND_CLR	R/W	0h	Number of corrected pending interrupts (same value as sec_pend). Writing decrements sec_pend by that value. Reset type: SYSRSn
7-6	INJ_UNCOR_PEND	R/W	0h	Number of injected uncorrected pending interrupts. Writing increments by that value Reset type: SYSRSn
5-4	INJ_COR_PEND	R/W	0h	Number of injected corrected pending interrupts. Writing increments by that value Reset type: SYSRSn
3-2	UNCOR_PEND	R/W	0h	Number of uncorrected pending interrupts. Writing increments by that value. Reset type: SYSRSn
1-0	COR_PEND	R/W	0h	Number of corrected pending interrupts. Writing increments by that value Reset type: SYSRSn

7.2.2.1.1.6 ERRORSTATUS2 Register (Offset = 24h) [Reset = 00000000h]

ERRORSTATUS2 is shown in Figure 7-7 and described in Table 7-8.

Return to the Summary Table.

Error Status 2 Register

Figure 7-7. ERRORSTATUS2 Register

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

Table 7-8. ERRORSTATUS2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ERROR_TYPE	R	0h	Error Type Reset type: SYSRSn
15-0	ERR_BIT	R	0h	Bit information that casued the error Reset type: SYSRSn

7.2.2.2 Read Operation on EDC Controller

User needs to follow the outlined steps for read transaction on serial VBUS :

1. Write Vector register with the appropriate vector field ID, address of EDC Controller register to rd_svbus_address, and set rd_svbus to a 1 to trigger a read on the serial VBUS.
2. Poll Vector register till rd_svbus_done (bit 24) is 1
3. Once rd_svbus_done = 1 do a normal read of the reserved_svbus_y register in the safety aggregator address space, this returns the read value

7.2.2.3 Write Operation on EDC Controller

User needs to follow the outlined steps for write transaction on serial VBUS :

1. Write the Vector register with the appropriate vector ID (bits 10-0), make sure rd_svbus = 0. Read address is a don't care
2. Do a regular write operation to the address that user wants to write to.

7.2.2.4 Safety Aggregator Error Injection

User needs to follow the outlined steps for parity error injection using serial VBUS :

1. Write the Vector register (offset - 0x8) with the index of the EDC controller to vector ID (bits 10-0), make sure rd_svbus = 0
2. Write to Error 1 register (offset - 0x18) of the EDC controller to program following :
   a. The GRP (group of checker to inject)
   b. The bit location thats is to be flipped
3. Write to the Control register (offset - 0x14) of the EDC controller to start the injection with the following :
   a. Make sure CHECK field (bit 1) is enabled
   b. Set force_se (bit 3) for single bit error to trigger parity error
4. When an interrupt from the safety aggregator is detected (assuming interrupts are enabled), software can read the raw interrupt status registers DED_STATUS_REG0 and check which ESM EDC controller has parity error.
5. To further find out information about the checker that detects the error, follow this sequence:
   a. Do a read operation from the ESM EDC Controller to read ERROR STATUS 1 register (offset - 0x20)
   b. From Error Status 1 register - user can check which specific checker reported the error and how many injected and non-injected errors are pending.
6. User can then do a Write operation to Error Status 1 register to clear injected error while exiting the parity error interrupt ISR and write the EOI register to acknowledge future error interrupts.

7.2.3 ESM Subsystem Integration View

The subsystem interacts with multiple blocks at the device level. The key ones being sources of errors, CPU's, Interrupt modules, System Control modules, and I/O pin interfaces. User can configure the controls and access various states of errors and actions. Subsystem consolidates various error sources and captures them to determine the right action based on user configurations. Figure 7-8 describes how the subsystem integrates at the device level in detail.

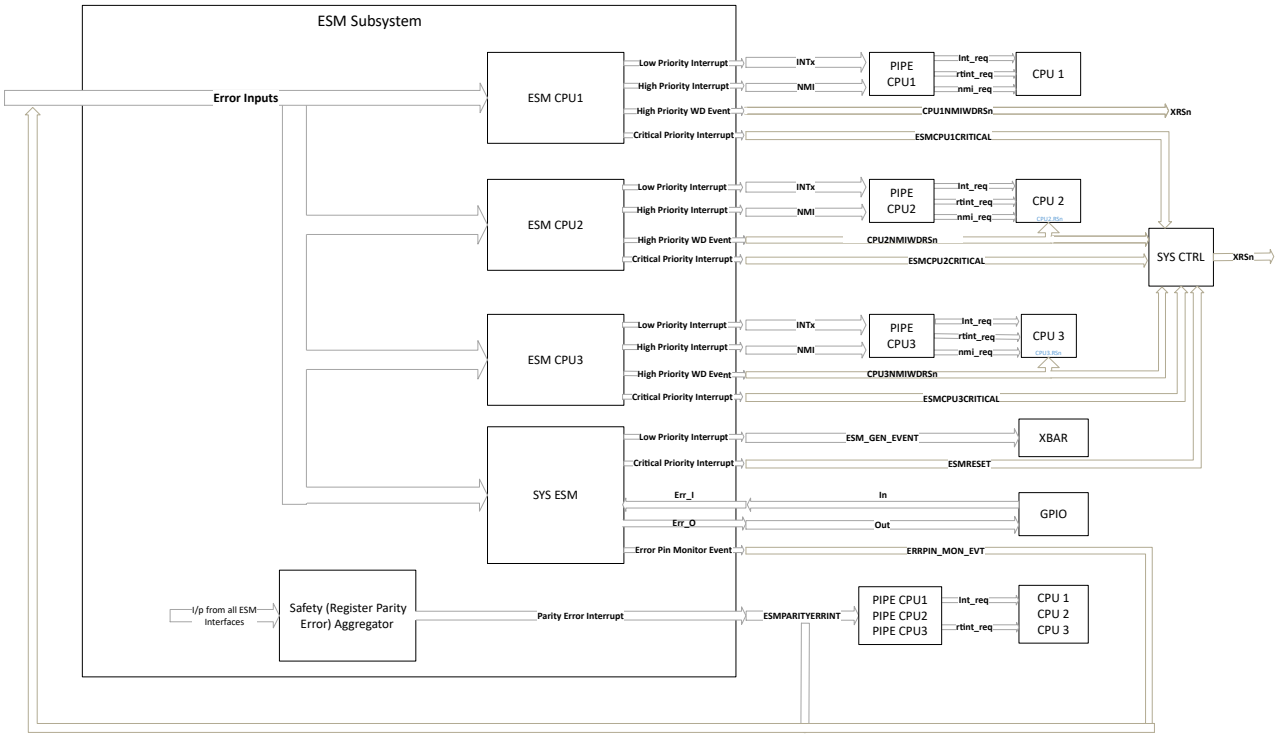


Figure 7-8. ESM Subsystem Integration View

7.3 ESM Functional Description

Figure 7-9 and Figure 7-10 illustrates the module features and functionality. The Error Signaling Module (ESM) centralizes fault reports and provides mechanisms to classify errors by severity and to provide programmable error response. The error classification in the ESM is determined by programmed configuration for each individual error input. For each individual error input the configuration can be set to assert an output error pin, or generate an interrupt to a CPU, or both. When an individual error input is configured to generate an interrupt, the configuration also selects whether the interrupt that is generated is high priority or low priority.

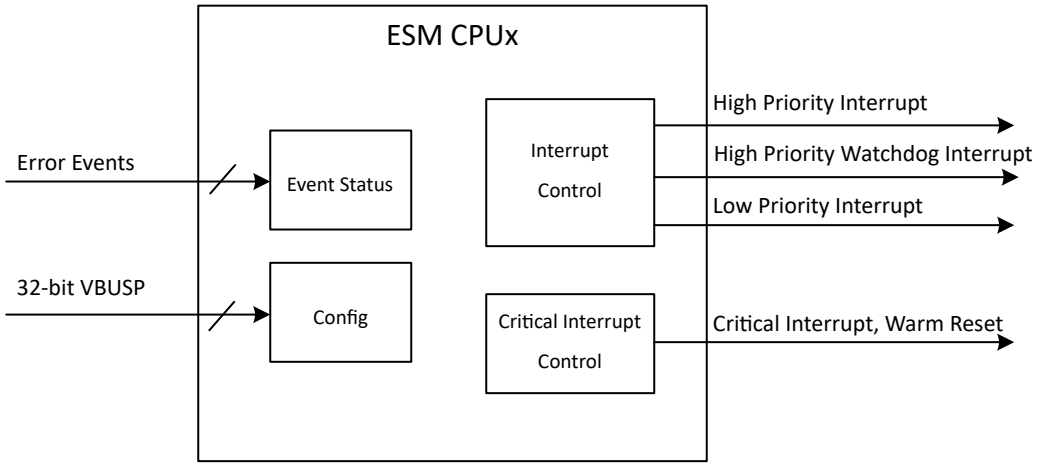


Figure 7-9. ESM CPU Block Diagram

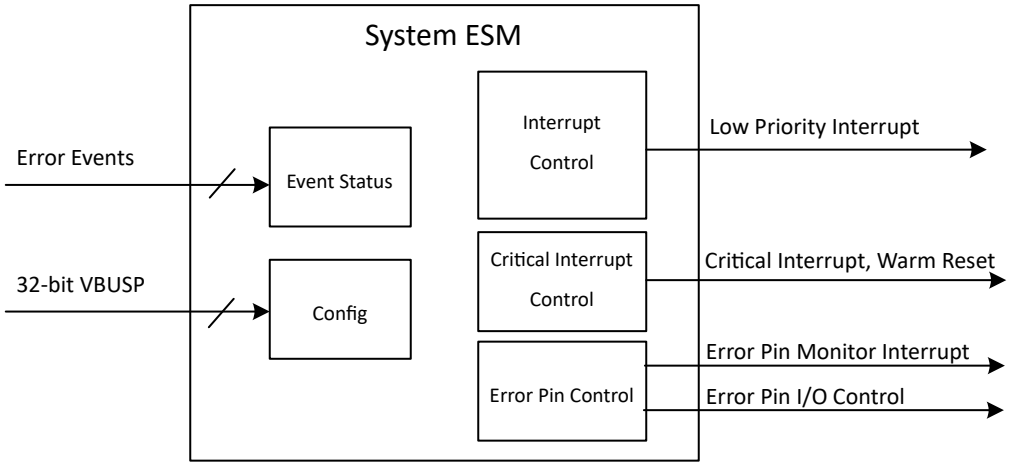


Figure 7-10. System ESM Block Diagram

By reporting the faults in a central location, the system can determine what caused the fault and what action can be taken. In general the faults can be split into two categories:

- Corrected faults
- Non-corrected faults

The ESM reports errors in two ways:

- An interrupt to a processor in the device. This allows the device to analyze and try to recover from an error.
- An external ERROR pin for System ESM module. This allows the system outside of the device to monitor for potentially fatal errors (errors that the device cannot self-recover from). Moreover, the external I/O (ERROR pin) can operate in level or PWM modes. In level mode, the output remains asserted (active low) for a minimum period of time. After that period of time, if the error has been cleared by an internal processor, the pin goes inactive (high). If signal does not go inactive in that time, then an external agent must intervene, as

an unrecoverable error can occur. In PWM mode, the error causes the output pin to maintain the value for a minimum period of time. After that period of time, if the error has been cleared by an internal processor, the pin continues the PWM pattern. If the signal does not go inactive in that time, then an external agent must intervene, as an unrecoverable error can occur.

Both mechanisms can be used at the same time for the same fault, signaling both an interrupt and the external ERROR pin. This allows the device to attempt to recover, but if recovery fails, then the external system is still alerted. If recovery succeeds, then the ERROR pin assertion can be removed so that the external system knows that a potentially unsafe condition was avoided.

Lastly, the ESM does not specify any methods of intervention, only the process of alerting internal CPUs and external monitors of an existing error event.

7.3.1 Error Event Inputs

Single ESM module supports up to 256 error event inputs, configurable by groups of 32. Error event inputs type can be either level or pulse. *N* (0-31) is the total number of groups, and *X* is the number of groups that are pulse. For this device, *N=X* means there are only pulse events and no level events. All events are relative to 0; hence, Event Group 0 Event 0 is global event 0 and sequentially Event Group1 Event 0 is global event 32.

Global event 0 to 31 are part of *Group0* similarly global event 32 to 63 belong to *Group1* and so on.

The pulse error event has a dedicated rising-edge detection circuit for each input.

Input Error Event Mapping to ESM is captured in Table 7-9.

Note

All Error Events that cause CPU to go into fault state must be configured to generate NMI (High Priority Interrupt) only. Group 0 events (Event Number 0 to 31) are mapped to High Priority Interrupt by default. CPUx_TMUROM_PAR_ERR also must to configured to generate NMI (high priority interrupt) in ESM configurations at device initialization.

All error event inputs are Pulse type only for this device.

Table 7-9. ESM Event Map

Event Number	Source Signal
0	ErrorAggregator_CPU1_HPERR
1	ErrorAggregator_CPU2_HPERR
2	ErrorAggregator_CPU3_HPERR
3	ErrorAggregator_CPU1_INT_HPERR
4	ErrorAggregator_CPU2_INT_HPERR
5	ErrorAggregator_CPU3_INT_HPERR
6-31	Reserved
32	CLOCKFAIL
33	DCC1_ERR
34	DCC2_ERR
35	DCC3_ERR
36	SYS_PLL_SLIP
37	CPU1RSn
38	CPU1WD
39	CPU1_ERAD_INT
40	CPU1_ERAD_NMI
41	ESM_CPU1_LOW_PRIORITY_INT
42	ESM_CPU1_HIGH_PRIORITY_INT

Table 7-9. ESM Event Map (continued)

Event Number	Source Signal
43	LCM_CPU1SS_LCMPERR
44	CPU2RSn
45	CPU2WD
46	CPU2_ERAD_INT
47	CPU2_ERAD_NMI
48	ESM_CPU2_LOW_PRIORITY_INT
49	ESM_CPU2_HIGH_PRIORITY_INT
50	CPU3RSn
51	CPU3WD
52	CPU3_ERAD_INT
53	CPU3_ERAD_NMI
54	ESM_CPU3_LOW_PRIORITY_INT
55	ESM_CPU3_HIGH_PRIORITY_INT
56	RTDMA_LCM_CMP_ERR
57	Reserved
58	ESM_ERRPIN_MON_EVT
59	ESM_PARITY_ERROR
60-62	Reserved
63	HSM_HEA_INT_LO
64	HSM_HEA_INT_HI
65	INPUTXBAR63
66	INPUTXBAR64
67	EPWMXBAR1
68	EPWMXBAR2
69	EPWMXBAR3
70	EPWMXBAR4
71	EPWMXBAR5
72	EPWMXBAR6
73	EPWMXBAR7
74	EPWMXBAR8
75	EPWMXBAR9
76	EPWMXBAR10
77	EPWMXBAR11
78	EPWMXBAR12
79	EPWMXBAR13
80	EPWMXBAR14
81	EPWMXBAR15
82	EPWMXBAR16
83	OUTPUTXBAR1
84	OUTPUTXBAR2
85	OUTPUTXBAR3
86	OUTPUTXBAR4
87	OUTPUTXBAR5
88	OUTPUTXBAR6
89	OUTPUTXBAR7

Table 7-9. ESM Event Map (continued)

Event Number	Source Signal
90	OUTPUTXBAR8
91	OUTPUTXBAR9
92	OUTPUTXBAR10
93	OUTPUTXBAR11
94	OUTPUTXBAR12
95	OUTPUTXBAR13
96	OUTPUTXBAR14
97	OUTPUTXBAR15
98	OUTPUTXBAR16
99	WADI1_INTN_O
100	WADI2_INTN_O
101	Reserved
102	CLB1_NMI
103	CLB2_NMI
104	CLB3_NMI
105	CLB4_NMI
106	CLB5_NMI
107	CLB6_NMI
108	EPG_INT
109	ECAT_NMIn
110	MCANA_ECC_CORR_PLS
111	MCANA_ECC_UNCORR_TS_PLS
112	MCANB_ECC_CORR_PLS
113	MCANB_ECC_UNCORR_TS_PLS
114	MCANC_ECC_CORR_PLS
115	MCANC_ECC_UNCORR_TS_PLS
116	MCAND_ECC_CORR_PLS
117	MCAND_ECC_UNCORR_TS_PLS
118	MCANE_ECC_CORR_PLS
119	MCANE_ECC_UNCORR_TS_PLS
120	MCANF_ECC_CORR_PLS
121	MCANF_ECC_UNCORR_TS_PLS
122	EMIF1_ERR
123	ADC_SAFETY_CHECK_INT_CPU1
124	ADC_SAFETY_CHECK_INT_CPU2
125	ADC_SAFETY_CHECK_INT_CPU3
126-127	Reserved
128	ErrorAggregator_CPU1_LPERR
129	Reserved
130	ErrorAggregator_CPU2_LPERR
131	Reserved
132	ErrorAggregator_CPU3_LPERR
133	Reserved
134	ErrorAggregator_CPU1_INT_LPERR
135	Reserved

Table 7-9. ESM Event Map (continued)

Event Number	Source Signal
136	ErrorAggregator_CPU2_INT_LPERR
137	Reserved
138	ErrorAggregator_CPU3_INT_LPERR
139	ErrorAggregator_RTDMA1_HPERR
140	ErrorAggregator_RTDMA1_LPERR
141	ErrorAggregator_RTDMA2_HPERR
142	ErrorAggregator_RTDMA2_LPERR
143	ErrorAggregator_SSU_HPERR
144-145	Reserved
146	ErrorAggregator_ECAT_MEM_HPERR
147	ErrorAggregator_ECAT_MEM_LPERR
148-161	Reserved
162	MCANA_FEVT0
163	MCANA_FEVT1
164	MCANA_FEVT2
165	MCANB_FEVT0
166	MCANB_FEVT1
167	MCANB_FEVT2
168	MCANC_FEVT0
169	MCANC_FEVT1
170	MCANC_FEVT2
171	MCAND_FEVT0
172	MCAND_FEVT1
173	MCAND_FEVT2
174	MCANE_FEVT0
175	MCANE_FEVT1
176	MCANE_FEVT2
177	MCANF_FEVT0
178	MCANF_FEVT1
179	MCANF_FEVT2
180	WADI1_BLOCK1_SIG1ERROR_BUS_O[0]
181	WADI1_BLOCK1_SIG2ERROR_BUS_O[1]
182	WADI1_BLOCK1_SIG_TO_SIGERROR_BUS_O[2]
183	WADI1_BLOCK2_SIG1ERROR_BUS_O[0]
184	WADI1_BLOCK2_SIG2ERROR_BUS_O[1]
185	WADI1_BLOCK2_SIG_TO_SIGERROR_BUS_O[2]
186	WADI1_BLOCK3_SIG1ERROR_BUS_O[0]
187	WADI1_BLOCK3_SIG2ERROR_BUS_O[1]
188	WADI1_BLOCK3_SIG_TO_SIGERROR_BUS_O[2]
189	WADI1_BLOCK4_SIG1ERROR_BUS_O[0]
190	WADI1_BLOCK4_SIG2ERROR_BUS_O[1]
191	WADI1_BLOCK4_SIG_TO_SIGERROR_BUS_O[2]
192	WADI2_BLOCK1_SIG1ERROR_BUS_O[0]
193	WADI2_BLOCK1_SIG2ERROR_BUS_O[1]
194	WADI2_BLOCK1_SIG_TO_SIGERROR_BUS_O[2]

Table 7-9. ESM Event Map (continued)

Event Number	Source Signal
195	WADI2_BLOCK2_SIG1ERROR_BUS_O[0]
196	WADI2_BLOCK2_SIG2ERROR_BUS_O[1]
197	WADI2_BLOCK2_SIG_TO_SIGERROR_BUS_O[2]
198	WADI2_BLOCK3_SIG1ERROR_BUS_O[0]
199	WADI2_BLOCK3_SIG2ERROR_BUS_O[1]
200	WADI2_BLOCK3_SIG_TO_SIGERROR_BUS_O[2]
201	WADI2_BLOCK4_SIG1ERROR_BUS_O[0]
202	WADI2_BLOCK4_SIG2ERROR_BUS_O[1]
203	WADI2_BLOCK4_SIG_TO_SIGERROR_BUS_O[2]
204-232	Reserved
233	EQEPERR
234	I2C_DIAG_EVENT
235	CPU1_OVFINT
236	CPU1_UVFINT
237	CPU1_DOVINT
238	CPU2_OVFINT
239	CPU2_UVFINT
240	CPU2_DOVINT
241	CPU3_OVFINT
242	CPU3_UVFINT
243	CPU3_DOVINT
244	MEMSS_REG_PAR_ERR
245	FRI_REG_PAR_ERR
246	SYSCTL_OR_ANALOGSYSCTL_REG_PAR_ERR
247	LCM_CPU1SS_REG_PAR_ERR
248	LCM_RTDMA_REG_PAR_ERR
249	WADI_REG_PAR_ERR
250	CPU1_TMUROM_PAR_ERR
251	CPU2_TMUROM_PAR_ERR
252	CPU3_TMUROM_PAR_ERR
253-255	Reserved

7.3.2 Error Interrupt Outputs

Each module generates Interrupt Output so that the processor in the device is signalled to intervene when an Error Event occurs. Each error event input is to be enabled to cause an Error Interrupt to occur (via Error Group *N* Interrupt Enabled Set Register). Additionally, each error event input is configured to be programmed to influence either the Low Priority (default) interrupt or the High Priority interrupt (via Error Group *N* Interrupt Priority Register). The Low Priority interrupt is intended for events that are of interest, but does not require immediate intervention. For example, an indication that there was a single bit error that was corrected can be configured to signal a low priority interrupt, so that information can be collected for statistical purposes. A High Priority interrupt is intended for events that need immediate attention. For example, an indication that there was an uncorrected two-bit error can be configured to signal a high priority interrupt.

7.3.2.1 High Priority Watchdog

A High Priority Watchdog can be enabled to monitor if the process associated with servicing the High Priority Interrupt has failed to do so. Count down starts from a Pre-Load value and triggers the High Priority Watchdog

Interrupt when the time has expired. As long as at least one enabled error associated with the High Priority Interrupt is outstanding, the Watchdog continues to count down.

The Watchdog only triggers the High Priority Watchdog Interrupt a single time until and unless the Interrupt is cleared. The High Priority Interrupt can be cleared with the High Priority Watchdog Interrupt Status/Clear Register or by writing the EOI vector to the EOI Interrupt Register.

If the Watchdog value is non-zero and all High Priority errors are cleared, the Watchdog timer resets to the Pre-Load value and waits until the next High Priority error to begin counting down again.

An MMR write to the Pre-Load value overwrites the Watchdog counter value if the Watchdog is inactive. If the Watchdog is counting down, a write to the Pre-Load value does not have any effect until the Watchdog is once again inactive.

If the Pre-Load value is zero, a High Priority Watchdog Interrupt is triggered when an error associated with the High Priority Interrupt occurs.

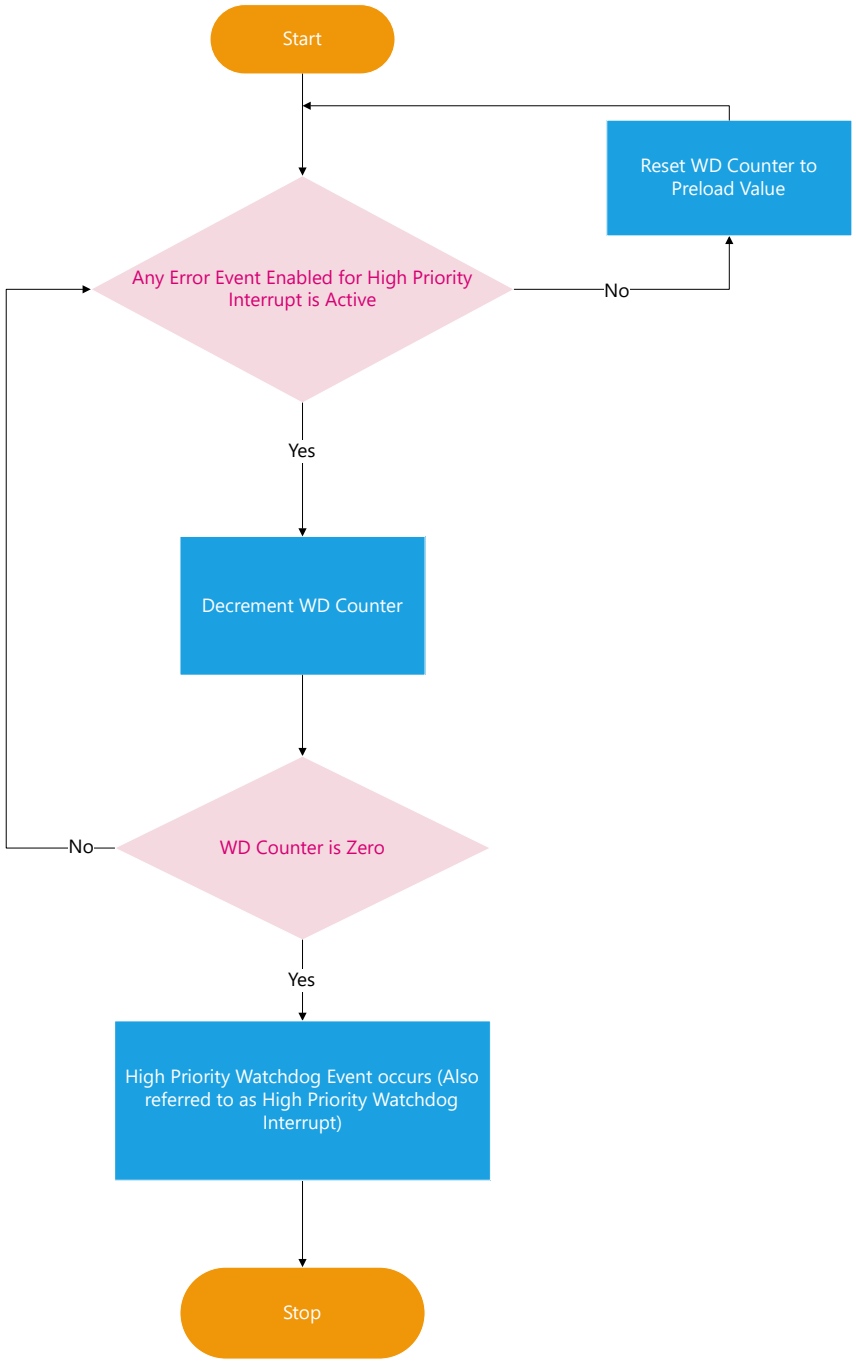


Figure 7-11. High Priority Watchdog Event Flowchart

7.3.2.2 Critical Priority Interrupt Output

The ESM supports a third interrupt output for events which are deemed even more critical than high priority events. For example, an error whose desired end result is a warm reset of a domain or the entire device. Unlike Low/High Priority Error Events, Critical Priority Error Events don't require enabling via the Error Group *N* Interrupt Enabled Set Register . Instead each Critical Priority Error Event Input is configured individually to influence the Critical Priority Interrupt Output via Error Group *N* Critical Priority Interrupt Influence Set Register .

The critical priority interrupt output is sensitive to the ESM's warm reset input as well as the Global Soft Reset MMR. The ESM's warm reset input is used synchronously and assertion of the ESM Warm Reset Input disables the ESM Global Enable MMR. When the critical priority interrupt output triggers, the occurrence is logged in the Info Register. The logging status is only reset via POR and the Global Soft Reset MMR.

7.3.3 Error Pin Output (ERR_O/ERRORSTS)

The error pin output (ERR_O/ERRORSTS) is used to signal an external agent to intervene because of an error. Each error event input can be programmed, via software, to influence the error pin output (using the Error Group *N* Error Pin Influence Set Register). The error pin output is active low or PWM based on the Error Pin Control Register[7-4] PWM_EN field. This bit field can only be modified when the ESM is disabled, based on the Global Enable register.

During Power-On Reset (POR), the error pin is active (asserted low) and the device drives this using a weak internal pull-down. The I/O is under the control of the device. When POR is removed from the ESM, error pin is driven by internal pull-down so the device can hand over control to the ESM. The user can also add an external pull-down that is only active when the device is in reset.

During a warm reset the state of the error pin is unchanged, that is the error pin logic is only reset by a POR. The device leaves the I/O active during a warm reset.

Figure 7-12 describes the behavior of the Error Pin. Not shown is that a reset (Power-On-Reset only) immediately transitions the Error pin to the ESM_RESET state and a Global Soft Reset immediately transitions the Error pin to the ESM_IDLE state. A Pending Error Event is any error event with the raw state set and the Error Pin Influence enabled. There are two types of "clear" events associated with servicing the Error Pin. The first is to clear the status of the pending event (see Section 7.5 for how to clear level and pulse pending events). The second is the CLEAR event meant to de-assert the Error Pin.

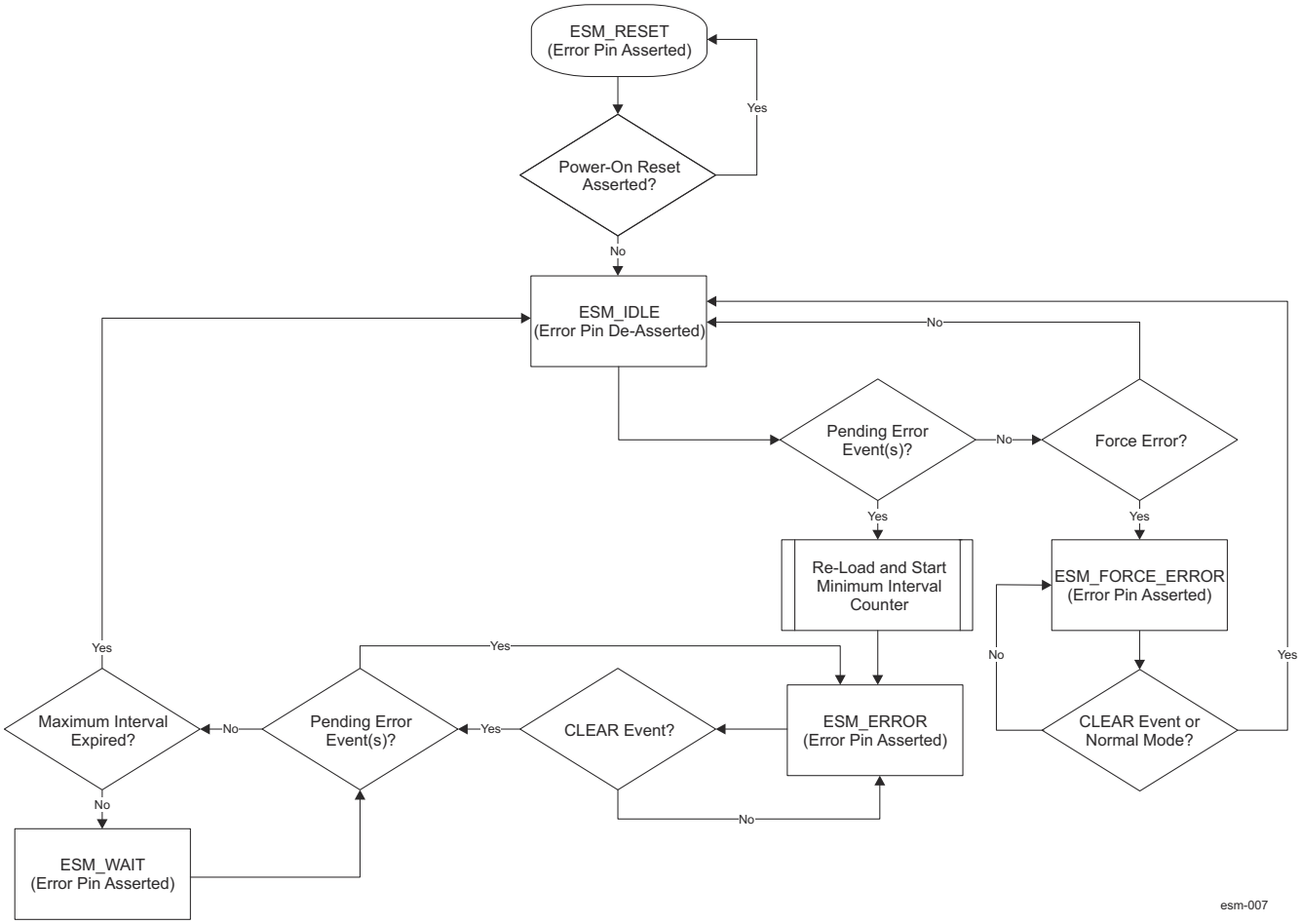


Figure 7-12. Error Pin State Flowchart

If an error event happens that has been programmed to influence the Error Pin, the Error Pin asserts (active low) for a minimum time as programmed (using the Error Pin Counter Pre-Load Register). For the Error Pin to de-assert, the following 3 things must happen : 1. The Minimum Time Interval must expire 2. The event that caused the Error Pin to assert must be cleared (see section for Interrupt Handling) 3. A CLEAR must be written to the Error Pin Control Register. Note that step 3 can only happen after step 2, but either (or both) of those steps can occur before or after step 1.

Figure 7-13 shows a typical error pin assertion.

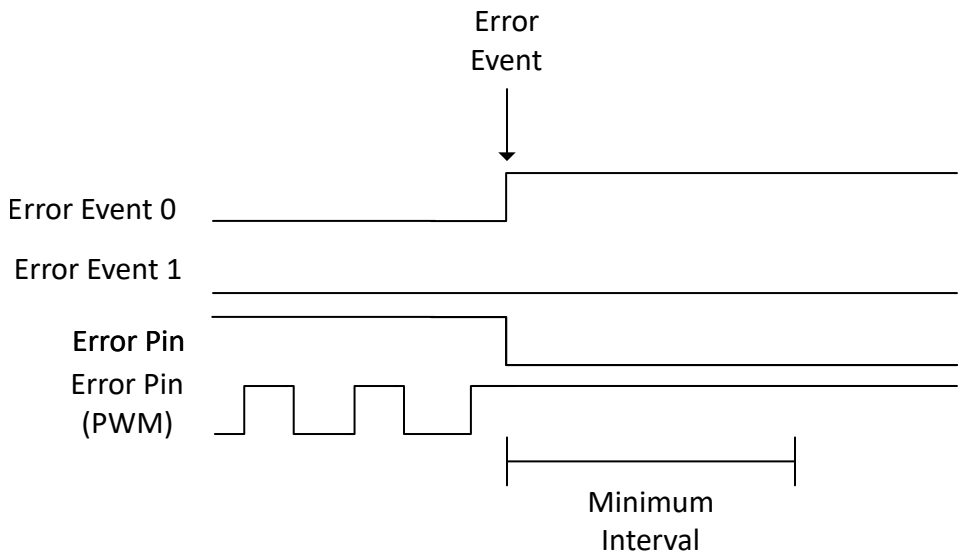


Figure 7-13. ESM Error Pin Assertion

If, during the minimum time, CLEAR is written to the error key, then the error pin de-asserts after the minimum time interval, as shown in Figure 7-14.

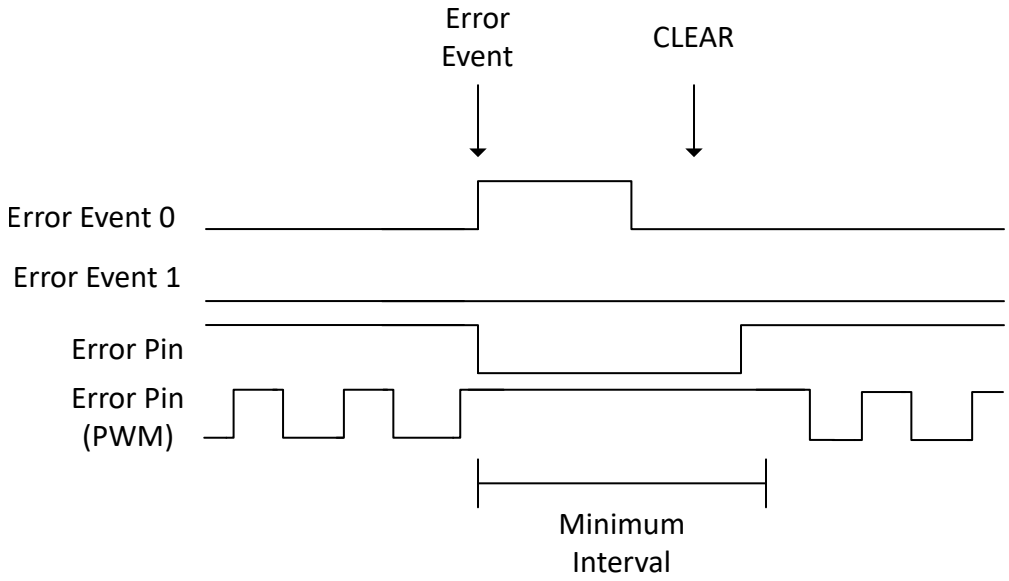


Figure 7-14. ESM Error Pin Assertion with CLEAR During Minimum Interval

If CLEAR is not written until after the Minimum Interval, the Error Pin de-asserts when CLEAR is written. This is regardless of whether the Error Event is removed before or after the Minimum Interval (as shown by the dotted line in Figure 7-15).

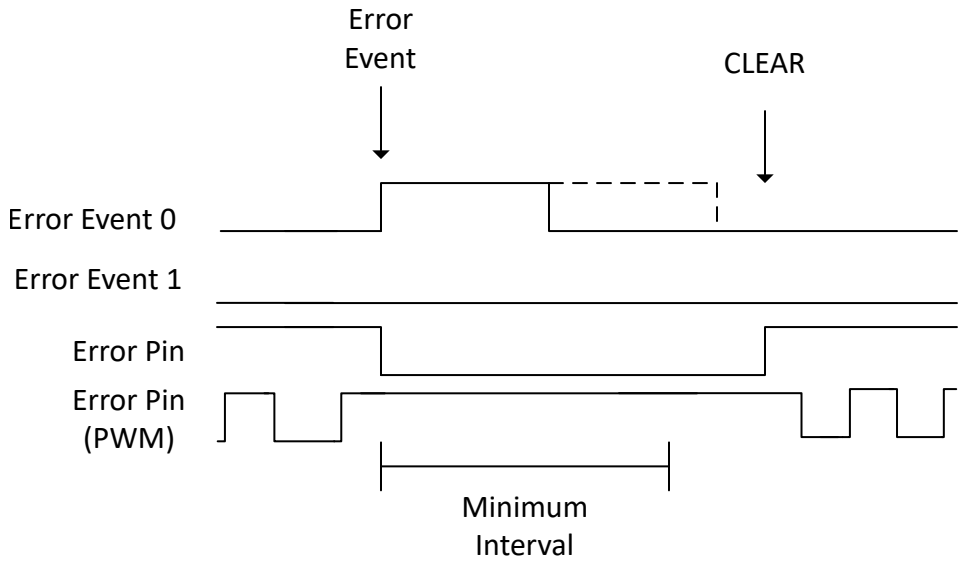


Figure 7-15. Error Pin Asserting with CLEAR after Minimum Interval

When in the ESM_ERROR state and a CLEAR event happens, if there are still pending Error Events, the ESM stays in the ESM_ERROR state with the Error Pin asserted. Multiple Error Events when in the ESM_ERROR state do not reset the Minimum Interval counter.

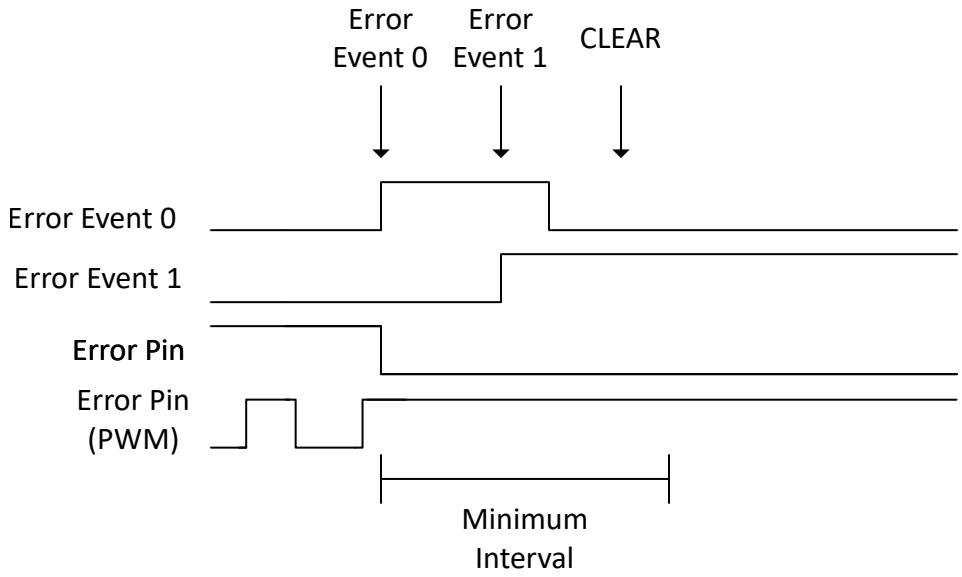


Figure 7-16. Error Pin Asserting with Interval Reset by Additional Error Event

A CLEAR event causes a re-evaluation of whether there are any pending Error Events. As such, a single CLEAR can be used to clear the Error Pin after multiple Error Events. Multiple CLEAR events can occur (such as the one with the dotted arrow in Figure 7-17), but are not necessary. No matter how many Error Events occur nor when (or how many) CLEAR events occur, the Error Pin is always asserted for at least the Minimum Interval.

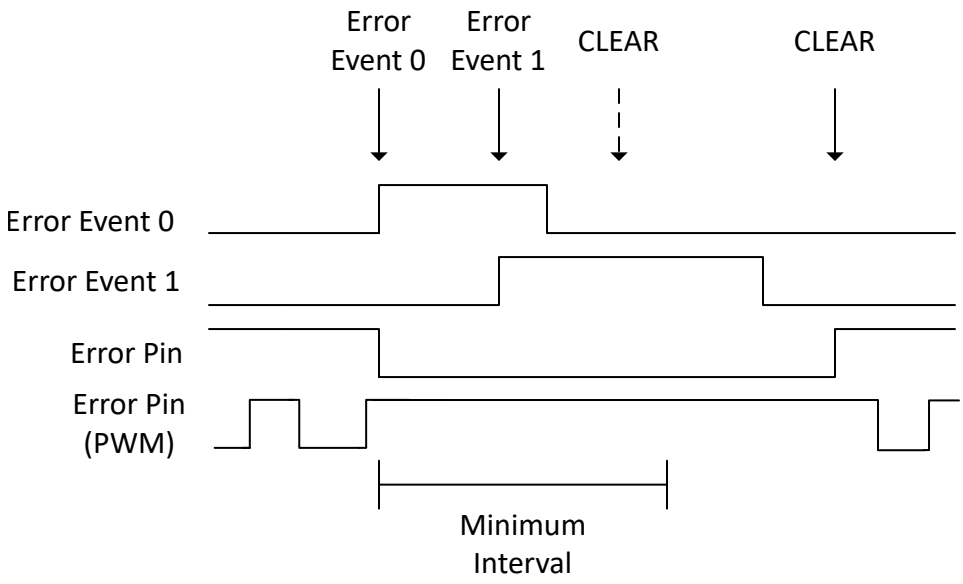


Figure 7-17. Error Pin Asserting with Single CLEAR for Multiple Events

If all Error Events are cleared and the ESM is in the ESM_WAIT state, waiting for the Minimum Interval to expire, and a new Error Event occurs, the ESM goes back to the ESM_ERROR state. The Minimum Interval does not reset, but a new CLEAR event is required.

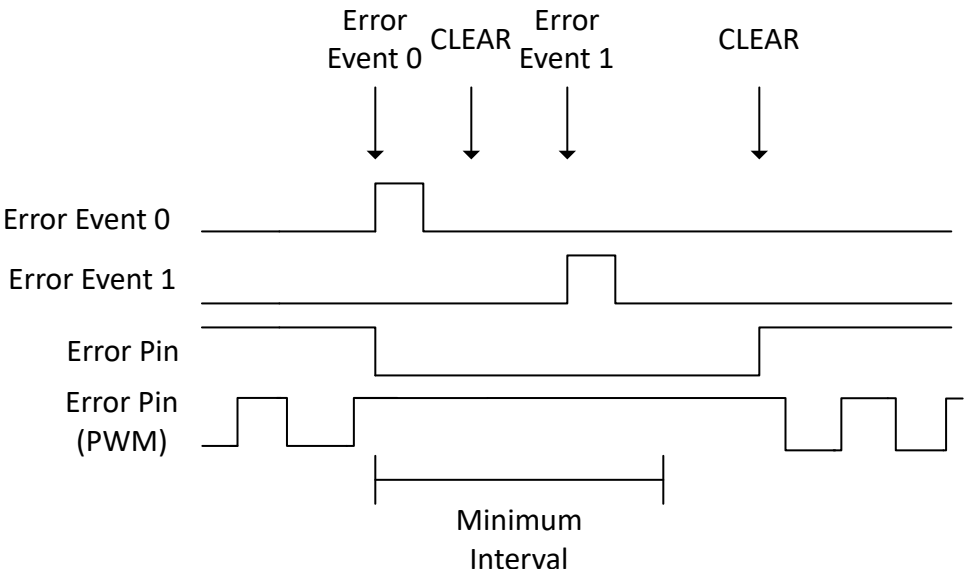


Figure 7-18. Error Pin Asserting with New Error During Minimum Time Interval

Table 7-10 shows some common scenarios of how the error pin as well as the two associated registers (Error Pin Control Register and Error Pin Status Register) are set.

Table 7-10. ESM Error Pin Scenarios

Scenario	Error Pin State Value	KEY	Error Pin Register Status	Additional Notes
POR Asserted	0	N/A	N/A	Registers are inaccessible. Device disables the I/O and pulls down internally.
After de-assertion of POR	1	0x0 (Normal Mode)	0x1	-
After de-assertion of Warm Reset (error was not asserted when reset asserted)	1	0x0 (Normal Mode)	0x1	-
After de-assertion of Warm Reset (error was asserted when reset asserted)	0	0x0 (Normal Mode)	0x0	-
Force error pin	0	0xA (Force Error Mode)	0x0	Forcing error on the pin using software.

7.3.3.1 Minimum Time Interval

The Minimum Time Interval is the minimum amount of time that the Error Pin remains asserted (active low) when an enabled Error Event occurs. This value is system dependent, but time interval must be high enough so that the external monitoring agent can always detect the Error Pin asserted, but short enough so that if all of the Error Events are cleared, then the Error Pin can be de-asserted before the external agent decides to intervene. This is highly dependent on the application and the Fault Tolerant Time Interval.

The Minimum Time Interval counter is clock cycle based, therefore the time of the interval is a combination of the value in the Error Pin Counter Pre-Load Register and the clock frequency of the ESM. For Software configuration the value must be calculated accordingly. The Minimum Time Interval default value is set to 0xFF FFFF (for both modes: level and pwm (high and low minimum time interval values)), although this value can be changed according to the needs of the application after start-up.

7.3.3.2 PWM Mode

If the error output pin is in PWM mode and when no error is detected pin toggles according to programmable MMR widths for high and low periods. When an error occurs, the error pin stops toggling and remains constant until the error is cleared. An external Power Management Integrated Circuits (PMIC) that is detecting the PWM toggles can identify the error if the pin stops toggling. The periods must be programmed considering the expectation of the external PMIC.

7.3.4 Reset Type Information for ESM Registers

Table 7-11. Reset Type Information

Register	Reset Domains		
	Module Reset	SFT_RST (Global Soft Reset)	PORESETn
INFO Register Bit 31			Yes
INFO Register Bit 30		Yes	Yes
EN (Global Enable)	Yes	Yes	Yes
LOW_PRI, HI_PRI, LOW, HI	Yes	Yes	Yes
HI_PRI_WD_CFG, HI_PRI_WD_CNTR, HI_PRI_WD_CNTR_PRE, HI_PRI_WD_INTR_SET, HI_PRI_WD_INTR_CLR	Yes	Yes	Yes
GROUP_N_LOCK, GROUP_N_COMMIT, ERR_PIN_INFLUENCE_LOCK, ERR_PIN_INFLUENCE_COMMIT, CRI_PRI_INFLUENCE_LOCK, CRI_PRI_INFLUENCE_COMMIT, MMR_CONFIG_LOCK, MMR_CONFIG_COMMIT	Yes	Yes	Yes
RAW_j , STS_j , INTR_EN_SET_j, INTR_EN_CLR_j, INT_PRIO_j, CRIT_EN_SET_j, CRIT_EN_CLR_j		Yes	Yes

Table 7-11. Reset Type Information (continued)

Register	Reset Domains		
	Module Reset	SFT_RST (Global Soft Reset)	PORESETn
PIN_CTRL, PIN_STS, PIN_CNTR, PIN_CNTR_PRE, PWMH_PIN_CNTR, PWMH_PIN_CNTR_PRE, PWML_PIN_CNTR, PWML_PIN_CNTR_PRE, ERRPIN_MON_CFG, ERRPIN_MON_INTR_SET, ERRPIN_MON_INTR_CLR		Yes	Yes
PIN_EN_SET_j, PIN_EN_CLR_j		Yes	Yes

Table 7-12. Module Reset Domain Information

ESM Subsystem Instance	Module Reset
ESM CPU1	CPU1SYSRSn
ESM CPU2	CPU2SYSRSn
ESM CPU3	CPU3SYSRSn
SYS ESM	XRSn

ESMCPU2 and ESMCPU3 instance only available after the respective CPU2 and CPU3 reset is released. If CPU2 is held in reset ESMCPU2 is not functional and same applies to ESMCPU3. Even when CPU2 is in lockstep with CPU1, ESMCPU2 instance is not available. If user wants to operate CPU2 as standalone core, lockstep needs to be disabled and then CPU2 reset needs to be released for ESMCPU2 instance to be functional. Table 7-12 helps to explain the above.

SYS ESM and Safety Aggregator instances is always available for functional use case irrespective of CPU2 or CPU3 being held in reset.

7.3.5 Clock Stop

The ESM can only be put in to clock stop if all of the internal state machines are idle and the global_enable is cleared (using the Global Enable register).

7.3.6 Commit/Lock for MMRs

Lock/Commit feature allows the configured Enable bits for given errors to be locked to prevent further changes. To prevent changes, the Lock must be written, then the Commit. The Commit prevents the Lock from changing. Both Commit and Lock MMRs are reset by the warm reset (module reset) and by global soft reset.

When Lock is set to 1, the associated MMR values can not be changed. When Commit is set to 1, the Lock value can not be changed. Additionally, once set to 1, the Commit value can not be changed. Both Commit and Lock are reset by the warm reset (mod_g_rst_n) and by global soft reset.

All of the writeable MMRs have Commit/Lock bits except for the following:

- EOI Interrupt Register
- All of Status/Clear Registers
  - Error Group *N* Event Raw Status/Set Register
  - Error Group *N* Event Enabled Status/Clear Register

7.3.7 Safety Protection for MMRs

The configuration MMRs are backed by parity checker, with one bit of parity per 8 bits of MMR data.

The parity detection logic is controlled by an EDC Controller instance, which can also be used to inject errors for testing purposes. The EDC Controller from each ESM module works with Safety Aggregator instance as described in Section 7.2.2.1.

7.3.8 Register Configuration Tieoffs

ESM Modules have MMR tieoffs namely Group 0 High priority Tieoffs and High Priority Watchdog Enable Tieoff set for the ESM Subsystem. Hence Group 0 events are mapped to High Priority Interrupt Outputs and High Priority Watchdog Counter is enabled by default.

7.3.8.1 Group0 High Priority Tieoff

All group 0 events are pre-set to be both enabled and high priority. This is the equivalent of a write to both the Error Group 0 Interrupt Enabled Set Register and Error Group 0 Interrupt Priority Register.

This setting defaults all Group 0 events to drive the high priority interrupt by default after POR or global soft reset. Group 0 events can still be configured. User can lock and commit by writing bit 0 of the Group *N* Interrupt Lock Register and Group *N* Interrupt Commit Register.

Additionally, pre-setting the error events for Group 0 in this way results in Group 0 by passing global enable. This means that a high priority interrupt can be triggered before software has written any configuration to the ESM.

7.3.8.2 High Priority Watchdog Enable Tieoff

The high priority watchdog enable is set which is equivalent to a write of 0xA to the High Priority Watchdog Config Register.

This defaults the High Priority Watchdog to be enabled by default after POR or global soft reset although user can still disabled by software. Additionally user can lock and commit the watchdog settings by writing to the watchdog lock/commit (bit 0) of the MMR Config Lock Register and MMR Config Commit Register. Pre-setting the High Priority Watchdog Enable in this way results in the High Priority Watchdog Interrupt bypassing global enable. This means that a high priority watchdog interrupt can be triggered before software has written any configuration to the ESM.

Note

As both High Priority Watchdog Enable Tieoff and the Group0 High Priority Enable Tieoff are enabled by default, care is taken that the high priority watchdog counter preload default value (0x0000 FFFF) is sufficiently large such that a Group 0 event that occurs during boot does not trigger a High Priority Watchdog Interrupt before a CPU can service the Group 0 event; as when the High Priority Watchdog triggers a CPU reset, this can result in a reset loop if the High Priority Watchdog triggers before a CPU is awake to handle a Group 0 event. User can change the preload counter value as required later.

7.4 ESM Configuration Guide

The following steps are to be performed by user depending on the error handling and condition control :

1. Configure the error event to affect either low priority or high priority interrupt for respective ESM instance. Error events that cause CPU to go into fault state must be configured to high priority interrupt to trigger NMI to respective CPU. For System ESM instance, high priority interrupt priority is not applicable.
2. Enable Interrupt generation for respective error event using Interrupt Enable Set Register. Based on Step1, the enabled interrupt event when active triggers selected priority error event.
3. Configure Global Enable to enable all interrupts, this is interrupt mask for all error events.
4. If the error event needs to affect the critical priority interrupt, set the respective event in the Critical Priority Interrupt Influence Set Register.
5. In Step1 if the interrupt priority is set to trigger high priority interrupt, watchdog on high priority interrupt is enabled at start-up with tie-off unless user wants to disable.
6. Optionally Lock and Commit the configurations.
7. For System ESM only : If error needs to be enabled to influence error pin, enable the event using Error Pin Influence Set Register. Error Pin configurations are only applicable to System ESM instance.
8. For System ESM only : After Step 7, Enable the Error Pin Monitor using Error Pin Monitor Config Register and enable Error Pin Monitor Interrupt using the Error Pin Monitor Interrupt Status/Set Register.

7.5 Interrupt Condition Control and Handling

This section describes in detail about each Interrupt type condition control and handling.

7.5.1 ESM Low Priority Error Interrupt

Any error event can be mapped to the low priority error interrupt. A low priority error interrupt is generated when an event is enabled to cause an interrupt (via the Error Group *N* Interrupt Enable Set Register) and mapped to the low priority error interrupt (via Error Group *N* Interrupt Priority Register) and the raw status is set (via the Error Group *N* Event Raw Status/Set Register).

When a Low Priority Error Interrupt is received, the acting processor is recommended to follow the outlined steps:

1. Read the Low Priority Prioritized Register
   - If both [31-16] PLS (low_pulse_prio) or [15-0] LVL (low_level_prio) bit fields are equal to 0xFFFF, then Interrupt is no longer asserted/pending.
   - If either [31-16] PLS (low_pulse_prio) or [15-0] LVL (low_level_prio) bit fields are not equal to 0xFFFF, software has two options for determining what event to service:
     – First Option: Record the value in low_pulse_prio and/or low_level_prio. Determine which is higher priority. This is the Global Event Number of the highest priority in Low Priority Error Event.
     – Second Option:
       - Read the Low Priority Interrupt Status Register to determine which Event Groups have pending Low Priority Interrupts.
       - Read the desired Error Group *N* Interrupt Enabled Status/Clear Register.
       - Identify which Low Priority Interrupt to service.
2. Based on the global event map for the device find the error event source.
3. Service the error event based on the individual peripheral specification:
   a. The system takes several actions including (but not limited to):
      i. Fixing the error
      ii. Resetting the errored peripheral
      iii. Resetting the device
      iv. Communicating outside the device via the error pin for outside intervention

Rest of the steps assume the error handling is complete and system wants to clear the error event. Clearing of error event is as described below :

Pulse Events

When a low priority error pulse event has to be cleared, the acting processor must perform the following steps in order:

1. Write 0x1 to the appropriate bit in the Error Group *N* Interrupt Enabled Status/Clear Register. This step clears the raw status and deassert the level interrupt.
2. Write the end of interrupt vector to the EOI Interrupt Register. In case there are enabled error events pending then a new pulse is generated and level interrupt remains asserted else no new pulse is generated.
3. Clear the Error Event at the source. The source generates a new pulse which shows up as a new error event at the ESM.
4. Write a CLEAR(0x5) to the Error Pin Control Register. This step is optional if the event is not enabled to influence the error pin (via the Error Pin Influence Set Register), but recommended to be done regardless as an extra CLEAR is not harmful.

7.5.2 ESM High Priority Error Interrupt

Any error event can be mapped to the high priority error interrupt. A high priority error interrupt is generated when an event is enabled to cause an interrupt (via the Error Group *N* Interrupt Enable Set Register) and mapped to the high priority error interrupt (via Error Group *N* Interrupt Priority Register) and the raw status is set (via the Error Group *N* Event Raw Status/Set Register).

When a high Priority Error Interrupt is received, the acting processor is recommended to follow the outlined steps:

1. Read the High Priority Prioritized Register
   - If both [31-16] PLS (high_pulse_prio) or [15-0] LVL (high_level_prio) bit fields are equal to 0xFFFF, then Interrupt is no longer asserted/pending.
   - If either [31-16] PLS (high_pulse_prio) or [15-0] LVL (high_level_prio) bit fields are not equal to 0xFFFF, software has two options for determining what event to service:
     – First Option: Record the value in high_pulse_prio and/or high_level_prio. Determine which is higher priority. This is the Global Event Number of the highest priority in high Priority Error Event.
     – Second Option:
       - Read the High Priority Interrupt Status Register to determine which Event Groups have pending high Priority Interrupts.
       - Read the desired Error Group *N* Interrupt Enabled Status/Clear Register.
       - Identify which high Priority Interrupt to service.
2. Determine, based on the global event map for the device find the error event source.
3. Service the error event based on the peripherals specification:
   a. The system takes several actions including (but not limited to):
      i. Fixing the error
      ii. Resetting the errored peripheral
      iii. Resetting the device
      iv. Communicating outside the device via the error pin for outside intervention

Rest of the steps assume the error handling is complete and system wants to clear the error event. Clearing of error event is as described below :

Pulse Events

When a high priority error pulse event has to be cleared, the acting processor must perform the following steps in order:

1. Write 0x1 to the appropriate bit in the Error Group *N* Interrupt Enabled Status/Clear Register. This step clears the raw status and deassert the level interrupt.
2. Write the end of interrupt vector to the EOI Interrupt Register. In case there are enabled error events pending then a new pulse is generated and level interrupt remains asserted else no new pulse is generated.
3. Clear the Error Event at the source. The source generates a new pulse which shows up as a new error event at the ESM.
4. Write a CLEAR(0x5) to the Error Pin Control Register. This step is optional if the event is not enabled to influence the error pin (via the Error Pin Influence Set Register), but recommended to be done regardless as an extra CLEAR is not harmful.

7.5.3 Critical Priority Error Interrupt

Any error event can be mapped to the critical priority interrupt. A high priority error interrupt is generated when an event is enabled to cause an interrupt (via the Error Group *N* Critical Priority Interrupt Influence Set Register) and the raw status is set (via the Error Group *N* Event Raw Status/Set Register).

When a critical priority error interrupt occurs following steps must be performed by acting processor :

1. Read the Info Register to confirm that the critical priority error output triggered.
2. Read Error Group *N* Critical Priority Interrupt Influence Set Register and the raw status in Error Group *N* Event Raw Status/Set Register to determine the input error events which triggered the critical priority error interrupt.
3. Perform Global Soft Reset to clear the Info Register (Base Address + 0x04) critical priority error interrupt status bit as well as all raw interrupt status
4. If the critical priority error interrupt resulted in an ESM Warm Reset, re-enable the Global Enable Register.

7.5.4 High Priority Watchdog Interrupt

The High Priority Watchdog Interrupt occurs when the process tasked with servicing the High Priority Interrupt has failed to do so for an amount of time set by the pre-loaded timeout value. The device can use the High Priority Watchdog Interrupt as a reset request.

Following actions must be taken by acting processor for Interrupt handling :

- If the High Priority Watchdog Interrupt triggers an ESM Warm Reset, Global Enable Register is reenabled after servicing all outstanding errors that triggered the High Priority Watchdog Interrupt and servicing the High Priority Interrupt.
- Write EOI vector to the EOI Interrupt Register or the Interrupt Clear results in a re-evaluation of the High Priority Watchdog Interrupt.

7.5.5 Safety Aggregator Interrupt Control and Handling

The safety aggregator generates the non-correctable error interrupt where hardware cannot correct the error in case of parity checker error.

The following is the sequence for servicing interrupts:

- Software enables the interrupts for a each edc_ctrl interface by writing 0x1 to the corresponding bit of the interrupt enable register (DED_ENABLE_SET_REG0).
- On receiving an interrupt, software checks which edc_ctrl interface has caused the error by reading the interrupt status (using DED_STATUS_SET_REG0).
- Software writes the edc_ctrl interface ID in the ECC vector field bits 10:0 along with the read message that includes setting bit 15 to trigger the serial VBUS read and writing the address to bits 23:16. The address corresponds to the register to read from the EDC controller component. Refer to the EDC Controller Register for more details.
- Software polls the read done bit (bit 24) in the ECC vector register.
- To clear the interrupt software does a write operation to Error Status 1 register to clear injected error and writes 0x1 to the end of interrupt register to clear the interrupt (using DED_EOI_REG).

7.6 Software

7.6.1 ESM_CPU Registers to Driverlib Functions

Table 7-13. ESM_CPU Registers to Driverlib Functions

File	Driverlib Function
PID	
-	
INFO	
esm.h	ESM_getLastResetSource
esm.h	ESM_getCriticalPriorityInterruptStatus
EN	
esm.h	ESM_enableGlobal
esm.h	ESM_disableGlobal
SFT_RST	
esm.h	ESM_triggerSoftReset
LOW_PRI	
esm.h	ESM_getHighestLowPriorityInterrupt
HI_PRI	
esm.c	ESM_setWatchdogCounterPreload
esm.h	ESM_enableHighPriorityWatchdog
esm.h	ESM_disableHighPriorityWatchdog
esm.h	ESM_enableHighPriorityWatchdogFreerun
esm.h	ESM_disableHighPriorityWatchdogFreerun
esm.h	ESM_getHighestHighPriorityInterrupt
LOW	
esm.h	ESM_getLowPriorityInterruptStatus
esm.h	ESM_getHighestLowPriorityInterrupt
HI	
esm.c	ESM_setWatchdogCounterPreload
esm.h	ESM_enableHighPriorityWatchdog
esm.h	ESM_disableHighPriorityWatchdog
esm.h	ESM_enableHighPriorityWatchdogFreerun
esm.h	ESM_disableHighPriorityWatchdogFreerun
esm.h	ESM_getHighPriorityInterruptStatus
esm.h	ESM_getHighestHighPriorityInterrupt
EOI	
esm.h	ESM_writeEOIVector
esm.h	ESM_ackInterrupt
PIN_CTRL	
-	
PIN_STS	
-	
PIN_CNTR	
-	
PIN_CNTR_PRE	
-	
PWMH_PIN_CNTR	

Table 7-13. ESM_CPU Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
PWMH_PIN_CNTR_PRE	
-	
PWML_PIN_CNTR	
-	
PWML_PIN_CNTR_PRE	
-	
HI_PRI_WD_CFG	
esm.h	ESM_enableHighPriorityWatchdog
esm.h	ESM_disableHighPriorityWatchdog
esm.h	ESM_enableHighPriorityWatchdogFreerun
esm.h	ESM_disableHighPriorityWatchdogFreerun
HI_PRI_WD_CNTR	
esm.c	ESM_setWatchdogCounterPreload
HI_PRI_WD_CNTR_PRE	
esm.c	ESM_setWatchdogCounterPreload
HI_PRI_WD_INTR_SET	
-	
HI_PRI_WD_INTR_CLR	
-	
GROUP_N_LOCK	
esm.h	ESM_lockErrorGroupInterruptConfig
esm.h	ESM_unlockErrorGroupInterruptConfig
GROUP_N_COMMIT	
esm.h	ESM_commitErrorGroupInterruptConfig
ERR_PIN_INFLUENCE_LOCK	
-	
ERR_PIN_INFLUENCE_COMMIT	
-	
CRI_PRI_INFLUENCE_LOCK	
esm.h	ESM_lockCriticalPriorityInterruptInfluenceConfig
esm.h	ESM_unlockCriticalPriorityInterruptInfluenceConfig
CRI_PRI_INFLUENCE_COMMIT	
esm.h	ESM_commitCriticalPriorityInterruptInfluenceConfig
MMR_CONFIG_LOCK	
esm.h	ESM_lockMMRConfig
esm.h	ESM_unlockMMRConfig
MMR_CONFIG_COMMIT	
esm.h	ESM_commitMMRConfig
RAW(i)	
esm.h	ESM_setRawInterruptStatus
esm.h	ESM_getRawInterruptStatus
STS(i)	
esm.h	ESM_getInterruptStatus
esm.h	ESM_clearRawInterruptStatus

Table 7-13. ESM_CPU Registers to Driverlib Functions (continued)

File	Driverlib Function
INTR_EN_SET(i)	
esm.c	ESM_configErrorEvent
esm.c	ESM_config
esm.h	ESM_enableInterrupt
INTR_EN_CLR(i)	
esm.h	ESM_disableInterrupt
INT_PRIO(i)	
esm.c	ESM_configErrorEvent
esm.c	ESM_config
esm.h	ESM_setInterruptPriorityLevel
PIN_EN_SET(i)	
esm.c	ESM_configErrorEvent
esm.c	ESM_config
esm.h	ESM_setInfluenceOnErrorPin
PIN_EN_CLR(i)	
esm.c	ESM_configErrorEvent
esm.h	ESM_setInfluenceOnErrorPin
CRIT_EN_SET(i)	
esm.c	ESM_configErrorEvent
esm.c	ESM_config
esm.h	ESM_setCriticalPriorityInterruptInfluence
CRIT_EN_CLR(i)	
esm.c	ESM_configErrorEvent
esm.h	ESM_setCriticalPriorityInterruptInfluence

7.6.2 ESM_SYS Registers to Driverlib Functions

Table 7-14. ESM_SYS Registers to Driverlib Functions

File	Driverlib Function
PID	
-	
INFO	
-	
EN	
-	
SFT_RST	
-	
LOW_PRI	
-	
HI_PRI	
-	
LOW	
-	
HI	
-	
EOI	

Table 7-14. ESM_SYS Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
PIN_CTRL	
esm.c	ESM_configErrorPin
esm.h	ESM_setErrorMode
esm.h	ESM_clearErrorPin
esm.h	ESM_setOutputPinMode
esm.h	ESM_setLevelModePolarity
PIN_STS	
esm.h	ESM_getErrorPinStatus
PIN_CNTR	
esm.c	ESM_setErrorPinCounterPreload
esm.h	ESM_getErrorPinCounterValue
PIN_CNTR_PRE	
esm.c	ESM_setErrorPinCounterPreload
PWMH_PIN_CNTR	
esm.c	ESM_setPWMCounterPreload
PWMH_PIN_CNTR_PRE	
esm.c	ESM_setPWMCounterPreload
PWML_PIN_CNTR	
esm.c	ESM_setPWMCounterPreload
PWML_PIN_CNTR_PRE	
esm.c	ESM_setPWMCounterPreload
ERRPIN_MON_CFG	
esm.c	ESM_configErrorPin
esm.h	ESM_enableErrorPinMonitor
esm.h	ESM_disableErrorPinMonitor
ERRPIN_MON_INTR_SET	
esm.h	ESM_setErrorPinMonitorInterrupt
esm.h	ESM_getErrorPinMonitorInterruptStatus
ERRPIN_MON_INTR_CLR	
esm.h	ESM_clearErrorPinMonitorInterrupt
GROUP_N_LOCK	
-	
GROUP_N_COMMIT	
-	
ERR_PIN_INFLUENCE_LOCK	
esm.h	ESM_lockErrorPinInfluenceConfig
esm.h	ESM_unlockErrorPinInfluenceConfig
ERR_PIN_INFLUENCE_COMMIT	
esm.h	ESM_commitErrorPinInfluenceConfig
CRI_PRI_INFLUENCE_LOCK	
-	
CRI_PRI_INFLUENCE_COMMIT	
-	
MMR_CONFIG_LOCK	

Table 7-14. ESM_SYS Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
MMR_CONFIG_COMMIT	
-	
RAW(i)	
-	
STS(i)	
-	
INTR_EN_SET(i)	
-	
INTR_EN_CLR(i)	
-	
INT_PRIO(i)	
-	
PIN_EN_SET(i)	
-	
PIN_EN_CLR(i)	
-	
CRIT_EN_SET(i)	
-	
CRIT_EN_CLR(i)	
-	

7.6.3 ESM_SAFETY_AGGREGATOR Registers to Driverlib Functions

Table 7-15. ESM_SAFETY_AGGREGATOR Registers to Driverlib Functions

File	Driverlib Function
AGGR_REVISION	
-	
ECC_VECTOR	
-	
MISC_STATUS	
-	
DED_EOI_REG	
-	
DED_STATUS_REG0	
-	
DED_ENABLE_SET_REG0	
-	
DED_ENABLE_CLR_REG0	
-	
AGGR_ENABLE_SET	
-	
AGGR_ENABLE_CLR	
-	
AGGR_STATUS_SET	
-	

Table 7-15. ESM_SAFETY_AGGREGATOR Registers to Driverlib Functions (continued)

File	Driverlib Function
AGGR_STATUS_CLR	
-	

7.6.4 ESM Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/esm

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

7.6.4.1 ESM Multicore Example (CPU1,CPU3) - MULTI_CORE

FILE: esm_cpu1_cpu3_multi_c29x1.c

This example demonstrates how to handle fault conditions using ESM and Error Aggregator in a muli-core example (ESM_cpu1_cpu3_multi_c29x1.c).

When using CCS for debugging this Multi-core example, after launching the debug session,

- Connect to CPU1 and load only the c29x1.out.
- After the program is loaded, run CPU1.
- Once c29x1 configures and releases CPU3 out of reset, the program stops.
- Connect to CPU3 target now. c29x3.out would have started execution as soon as it is released from reset.
- In case of RAM configuration, restart the CPU3 target and load the symbols.

For FLASH configuration, this example is run in FLASH BANKMODE2, where CPU3 has access to FLASH (FRI-2). Refer to the Flash Plugin documentation to know about changing FLASH BANKMODEs and more. Additionally, the CPAx and CDAx RAMs are used for allocating various RAM sections. *External Connections*

- None.

Watch Variables

- None.

7.6.4.2 ESM Multicore Example (CPU1,CPU3) - MULTI_CORE

FILE: esm_cpu1_cpu3_multi_c29x3.c

This example demonstrates how to handle fault conditions using ESM and Error Aggregator in a muli-core example (ESM_cpu1_cpu3_multi_c29x3.c).

When using CCS for debugging this Multi-core example, after launching the debug session,

- Connect to CPU1 and first load the c29x3.out. (When loaded to FLASH, this might not halt at main due to an error, since CPU1 cannot execute from CPU3 FLASH memory (FRI-2)).
- c29x1.out can be loaded next.
- After the program is loaded, run CPU1.
- Once c29x1 configures and releases CPU3 out of reset, the program stops.
- Connect to CPU3 target now. c29x3.out would have started execution as soon as it is released from reset.
- In case of RAM configuration, restart the CPU3 target and load the symbols.
- In case of FLASH, do a Power-on-Reset, connect to CPU1/ CPU3 and load the respective symbols.

For FLASH configuration, this example is run in FLASH BANKMODE2, where CPU3 has access to FLASH (FRI-2). Refer to the Flash Plugin documentation to know about changing FLASH BANKMODEs and more. Additionally, the CPAx and CDAx RAMs are used for allocating various RAM sections. *External Connections*

- None.

Watch Variables

- CPU3nmigen
- Globalerroreventnumber

7.6.4.3 ESM - SINGLE_CORE

FILE: esm_ex1_tmuromparity_test.c

Example shows how to generate and check the error response configuration during development phase of software. It highlights the capability to self test the error propogation path providing test for diagnostic (TMU ROM Parity logic) Note - TMU ROM parity error puts CPU in fault state and only NMI can take CPU out of the fault state.

External Connections

- None.

Watch Variables

- error_generated - check if the proper error is generated

7.6.4.4 ESM - SINGLE_CORE

FILE: esm_ex2_safetyaggregator.c

This example shows how to do parity error injection for safety aggregator and generate parity error interrupt using the safety aggregator. For parity error generation, the example injects a single bit error and waits for the code execution to jump to parity error ISR in for loop. The parity error ISR clears the DED_STATUS_REG and sets EOI to acknowledge future Parity interrupts.

External Connections

- None.

Watch Variables

- ISRCount - The number of times entered into the ESM Parity Error ISR
- loopCount - The number of loops performed while not in ISR

7.7 ESM Registers

This Section describes the ESM Registers.

7.7.1 ESM Base Address Table

Table 7-16. ESM Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
ESM_CPU_REGS	ESMCPU1_BASE	0x6009_0000	YES	YES	YES	YES	YES	YES	-	YES
ESM_CPU_REGS	ESMCPU2_BASE	0x6009_1000	YES	YES	YES	YES	YES	YES	-	YES
ESM_CPU_REGS	ESMCPU3_BASE	0x6009_2000	YES	YES	YES	YES	YES	YES	-	YES
ESM_SYSTEM_REGS	ESMSYSTEM_BASE	0x6009_F000	YES	YES	YES	YES	YES	YES	-	YES
ESM_SAFETYAGG_REGS	ESMSAFETYAGG_BASE	0x600A_0000	YES	YES	YES	YES	YES	YES	-	YES

7.7.2 ESM_CPU_REGS Registers

Table 7-17 lists the memory-mapped registers for the ESM_CPU_REGS registers. All register offset addresses not listed in Table 7-17 should be considered as reserved locations and the register contents should not be modified.

Table 7-17. ESM_CPU_REGS Registers

Offset	Acronym	Register Name
0h	PID	Revision Register
4h	INFO	Info Register
8h	EN	Global Enable Register
Ch	SFT_RST	Global Soft Reset Register
20h	LOW_PRI	Low Priority Prioritized Register
24h	HI_PRI	High Priority Prioritized Register
28h	LOW	Low Priority Interrupt Status Register
2Ch	HI	High Priority Interrupt Status Register
30h	EOI	EOI Interrupt Register
80h	HI_PRI_WD_CFG	High Priority Watchdog Config Register
84h	HI_PRI_WD_CNTR	High Priority Watchdog Counter Value Register
88h	HI_PRI_WD_CNTR_PRE	High Priority Watchdog Pre-Load Register
8Ch	HI_PRI_WD_INTR_SET	High Priority Watchdog Interrupt Status/Set Register
90h	HI_PRI_WD_INTR_CLR	High Priority Watchdog Interrupt Status/Clear Register
100h	GROUP_N_LOCK	Group N Interrupt Lock Register
104h	GROUP_N_COMMIT	Group N Interrupt Commit Register
118h	CRI_PRI_INFLUENCE_LOCK	Critical Prioirty Interrupt Influence Lock Register
11Ch	CRI_PRI_INFLUENCE_COMMIT	Critical Priority Interrupt Influence Lock Register
120h	MMR_CONFIG_LOCK	MMR Config Lock Register
124h	MMR_CONFIG_COMMIT	MMR Config Commit Register
400h + formula	RAW_j	Error Group N Event Raw Status/Set Register
404h + formula	STS_j	Error Group N Interrupt Enable Status/Clear Register
408h + formula	INTR_EN_SET_j	Error Group N Interrupt Enable Set Register
40Ch + formula	INTR_EN_CLR_j	Error Group N Interrupt Enabled Clear Register
410h + formula	INT_PRIO_j	Error Group N Interrupt Priority Register
800h + formula	CRIT_EN_SET_j	Error Group N Critical Priority Interrupt Influence Set Register
804h + formula	CRIT_EN_CLR_j	Error Group N Critical Priority Interrupt Influence Clear Register

Complex bit access types are encoded to fit into small table cells. Table 7-18 shows the codes that are used for access types in this section.

Table 7-18. ESM_CPU_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		

Table 7-18. ESM_CPU_REGS Access Type Codes (continued)

Access Type	Code	Description
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

7.7.2.1 PID Register (Offset = 0h) [Reset = 6FE03101h]

PID is shown in Figure 7-19 and described in Table 7-19.

Return to the Summary Table.

The Revision Register contains the major and minor revisions for the module.

Figure 7-19. PID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
scheme		bu		func											
R-1h		R-2h		R-FE0h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rtl					major			custom		minor					
R-6h					R-1h			R-0h		R-1h					

Table 7-19. PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	scheme	R	1h	PID register scheme
29-28	bu	R	2h	Business Unit: 10 = Processors
27-16	func	R	FE0h	Module ID
15-11	rtl	R	6h	RTL revision. Will vary depending on release.
10-8	major	R	1h	Major revision
7-6	custom	R	0h	Custom
5-0	minor	R	1h	Minor revision

7.7.2.2 INFO Register (Offset = 4h) [Reset = XXXX0808h]

INFO is shown in Figure 7-20 and described in Table 7-20.

Return to the Summary Table.

The Info Register gives the configuration Inforrmation of this ESM.

Figure 7-20. INFO Register

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

Table 7-20. INFO Register Field Descriptions

Bit	Field	Type	Reset	Description
31	last_reset	R	0h	Indicates the Source of the last Reset 0 – Last reset was a Power On Reset 1 – Last reset was a Warm Reset
30	crit_intr	R	0h	Indicates if the critical priority interrupt output has asserted 0 – Critical Priority Interrupt output has not triggered 1 – Critical Priority Interrupt output has triggered
29-16	RESERVED	R	XXXh	
15-8	pulse_groups	R	8h	Number of Pulse Error Groups
7-0	groups	R	8h	Total number of Error Groups

7.7.2.3 EN Register (Offset = 8h) [Reset = 00000000h]

EN is shown in Figure 7-21 and described in Table 7-21.

Return to the Summary Table.

The Global Enable Register has the master interrupt mask

Figure 7-21. EN Register

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

Table 7-21. EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	key	R/W	0h	Global Enable. 4'b0000- All interrupts are disabled. 4'b1111- All interrupts are enabled

7.7.2.4 SFT_RST Register (Offset = Ch) [Reset = 00000000h]

SFT_RST is shown in Figure 7-22 and described in Table 7-22.

Return to the Summary Table.

The Global Soft Reset Register controls the global clear for raw status and enables

Figure 7-22. SFT_RST Register

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

Table 7-22. SFT_RST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3-0	key	W	0h	Global Soft Reset 4'b1111- Clear all raw status and enable bits. All Others - No effect

7.7.2.5 LOW_PRI Register (Offset = 20h) [Reset = FFFFFFFFh]

LOW_PRI is shown in Figure 7-23 and described in Table 7-23.

Return to the Summary Table.

Shows which is the highest priority outstanding low priority interrupt

Figure 7-23. LOW_PRI Register

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

Table 7-23. LOW_PRI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	pls	R	FFFFh	This is the highest priority outstanding low priority pulse interrupt The lowest event number has the highest priority. A value of all ones (0xFFFF) indicates that there are no low priority interrupts pending. This field is updated whenever a new, higher priority event occurs
15-0	lvl	R	FFFFh	This is the highest priority outstanding low priority level interrupt The lowest event number has the highest priority. A value of all ones (0xFFFF) indicates that there are no low priority interrupts pending. This field is updated whenever a new, higher priority event occurs

7.7.2.6 HI_PRI Register (Offset = 24h) [Reset = FFFFFFFFh]

HI_PRI is shown in Figure 7-24 and described in Table 7-24.

Return to the Summary Table.

Shows which is the highest priority outstanding high priority interrupt

Figure 7-24. HI_PRI Register

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

Table 7-24. HI_PRI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	pls	R	FFFFh	This is the highest priority outstanding high priority pulse interrupt The lowest event number has the highest priority. A value of all ones (0xFFFF) indicates that there are no high priority interrupts pending. This field is updated whenever a new, higher priority event occurs
15-0	lvl	R	FFFFh	This is the highest priority outstanding high priority level interrupt The lowest event number has the highest priority. A value of all ones (0xFFFF) indicates that there are no high priority interrupts pending. This field is updated whenever a new, higher priority event occurs

7.7.2.7 LOW Register (Offset = 28h) [Reset = 00000000h]

LOW is shown in Figure 7-25 and described in Table 7-25.

Return to the Summary Table.

Shows which groups have oustanding low priority interrupts

Figure 7-25. LOW Register

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

Table 7-25. LOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	sts	R	0h	This is the raw status for low priority interrupt event groups Indicates which Event Groups have one or more Low Priority interrupts pending. This register is bit oriented where bit 0 is for Event Group 0, bit 1 is for Event Group 1, bit N is for Event Group N.

7.7.2.8 HI Register (Offset = 2Ch) [Reset = 00000000h]

HI is shown in Figure 7-26 and described in Table 7-26.

Return to the Summary Table.

Shows which groups have oustanding high priority interrupts

Figure 7-26. HI Register

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

Table 7-26. HI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	sts	R	0h	This is the raw status for high priority interrupt event groups Indicates which Event Groups have one or more High Priority interrupts pending. This register is bit oriented where bit 0 is for Event Group 0, bit 1 is for Event Group 1, bit N is for Event Group N.

7.7.2.9 EOI Register (Offset = 30h) [Reset = 00000000h]

EOI is shown in Figure 7-27 and described in Table 7-27.

Return to the Summary Table.

End of Interrupt Register

Figure 7-27. EOI Register

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

Table 7-27. EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	W	0h	
10-0	key	W	0h	This is the interrupt being serviced 1h = Low Priority Error Interrupt 2h = High Priority Error Interrupt 4h = High Priority Watchdog Interrupt

7.7.2.10 HI_PRI_WD_CFG Register (Offset = 80h) [Reset = 000000XAh]

HI_PRI_WD_CFG is shown in Figure 7-28 and described in Table 7-28.

Return to the Summary Table.

The High Priority Watchdog Config Register is used to enable or disable the High Priority Watchdog and its associated interrupt.

Figure 7-28. HI_PRI_WD_CFG Register

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

Table 7-28. HI_PRI_WD_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-8	emu_free	R/W	0h	Enable free running mode for the High Priority Watchdog to bypass the debug suspend input. 0xA Enable the watchdog counter during debug suspend. For all other values the counter will freeze during emulation when the debug suspend input is high.
7-4	RESERVED	R/W	Xh	
3-0	wd_en	R/W	Ah	Enable field for the High Priority Watchdog. 0xA Enables the watchdog. All other values disable the watchdog

7.7.2.11 HI_PRI_WD_CNTR Register (Offset = 84h) [Reset = 0000FFFFh]

HI_PRI_WD_CNTR is shown in Figure 7-29 and described in Table 7-29.

Return to the Summary Table.

The High Priority Watchdog Counter Register reflects the current value of the High Priority Watchdog Counter. When enabled, this counter will start decrementing (from the High Priority Pre-Load value) as soon as any of the enabled Errors associated with the High Priority Interrupt is active. This counter is reset by the warm reset.

Figure 7-29. HI_PRI_WD_CNTR Register

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

Table 7-29. HI_PRI_WD_CNTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	count	R	FFFFh	Current Counter Value

7.7.2.12 HI_PRI_WD_CNTR_PRE Register (Offset = 88h) [Reset = 0000FFFFh]

HI_PRI_WD_CNTR_PRE is shown in Figure 7-30 and described in Table 7-30.

Return to the Summary Table.

The High Priority Watchdog Pre-Load Register reflects the value that is loaded into the High Priority Watchdog Counter Register.

Figure 7-30. HI_PRI_WD_CNTR_PRE Register

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

Table 7-30. HI_PRI_WD_CNTR_PRE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	count	R/W	FFFFh	Counter Pre-Load Value

7.7.2.13 HI_PRI_WD_INTR_SET Register (Offset = 8Ch) [Reset = 00000000h]

HI_PRI_WD_INTR_SET is shown in Figure 7-31 and described in Table 7-31.

Return to the Summary Table.

The High Priority Watchdog Status/Set Register indicates the status of the High Priority Watchdog Interrupt. This register is reset by the Power On Reset. A write to this register may allow software to set the interrupt output.

Figure 7-31. HI_PRI_WD_INTR_SET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															val
R/W-0h															R/W1S-0h

Table 7-31. HI_PRI_WD_INTR_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	val	R/W1S	0h	Read the current interrupt status. Write 1 to set the interrupt output

7.7.2.14 HI_PRI_WD_INTR_CLR Register (Offset = 90h) [Reset = 00000000h]

HI_PRI_WD_INTR_CLR is shown in Figure 7-32 and described in Table 7-32.

Return to the Summary Table.

The High Priority Watchdog Status/Clear Register indicates the status of the High Priority Watchdog Interrupt. This register is reset by the Power On Reset. A write to this register may allow software to temporarily clear the interrupt.

Figure 7-32. HI_PRI_WD_INTR_CLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															val
R/W-0h															R/W1C-0h

Table 7-32. HI_PRI_WD_INTR_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	val	R/W1C	0h	Read the current interrupt status. Write 1 to clear the interrupt output. The interrupt will reassert if the conditions that triggered it persist

7.7.2.15 GROUP_N_LOCK Register (Offset = 100h) [Reset = 00000000h]

GROUP_N_LOCK is shown in Figure 7-33 and described in Table 7-33.

Return to the Summary Table.

The fields of the Group N Interrupt Lock Register lock the configuration for the associated Error Group N Interrupt Enable Set/Clear Registers and the Error Group N Interrupt Priority Register. This locks the associated enable and priority bits for each Group for Low and High Priority Interrupts. Locks may be written until the associated Commit MMR fields are set to 1. Once Lock is 1 and Commit is 1, the associated configuration may not be changed until reset.

Figure 7-33. GROUP_N_LOCK Register

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

Table 7-33. GROUP_N_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	lock	R/W	0h	Each bit lock[N] bit will lock the associated masking MMRs associated with Group N. These are: Error Group N Interrupt Enabled Set Register, Error Group N Interrupt Enabled Clear Register, Error Group N Interrupt Priority Register

7.7.2.16 GROUP_N_COMMIT Register (Offset = 104h) [Reset = 00000000h]

GROUP_N_COMMIT is shown in Figure 7-34 and described in Table 7-34.

Return to the Summary Table.

The fields of the Group N Interrupt Commit Register commit the lock configuration for the associated Group N Interrupt Lock Register. This prevents the Locks in the Group N Interrupt Lock Register from changing. Once Commit is 1, the associated Lock and Commit may not be changed until reset.

Figure 7-34. GROUP_N_COMMIT Register

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

Table 7-34. GROUP_N_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	commit	R/W	0h	Each bit commit[N] bit will commit the lock configuration for the corresponding bit in the Group N Interrupt Lock Register.

7.7.2.17 CRI_PRI_INFLUENCE_LOCK Register (Offset = 118h) [Reset = 00000000h]

CRI_PRI_INFLUENCE_LOCK is shown in Figure 7-35 and described in Table 7-35.

Return to the Summary Table.

The fields of the Critical Priority Interrupt Influence Lock Register lock the Critical Priority Interrupt Influence configuration for the associated Error Groups. Locks may be written until the associated Commit MMR fields are set to 1. Once Lock is 1 and Commit is 1, the associated configuration may not be changed until reset.

Figure 7-35. CRI_PRI_INFLUENCE_LOCK Register

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

Table 7-35. CRI_PRI_INFLUENCE_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	lock	R/W	0h	Each bit lock[N] bit will lock the associated masking MMRs associated with Group N. These are: Error Group N Critical Priority Interrupt Influence Set Register, Error Group N Critical Priority Interrupt Influence Clear Register

7.7.2.18 CRI_PRI_INFLUENCE_COMMIT Register (Offset = 11Ch) [Reset = 00000000h]

CRI_PRI_INFLUENCE_COMMIT is shown in Figure 7-36 and described in Table 7-36.

Return to the Summary Table.

The fields of the Critical Priority Interrupt Influence Commit Register commit the lock configuration for the associated Error Groups. This prevents the Locks in the Critical Priority Interrupt Influence Lock Register from changing. Once Commit is 1, the associated Lock and Commit may not be changed until reset.

Figure 7-36. CRI_PRI_INFLUENCE_COMMIT Register

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

Table 7-36. CRI_PRI_INFLUENCE_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	commit	R/W	0h	Each bit commit[N] bit will commit the lock configuration for the corresponding bit in the Critical Priority Interrupt Influence Lock Register.

7.7.2.19 MMR_CONFIG_LOCK Register (Offset = 120h) [Reset = 000000XXh]

MMR_CONFIG_LOCK is shown in Figure 7-37 and described in Table 7-37.

Return to the Summary Table.

The fields of the MMR Config Lock Register lock the configuration for the associated MMRs. This prevents changes once the lock configuration is committed with the MMR Config Commit Register. Locks may be written until the associated Commit MMR fields are set to 1. Once Lock is 1 and Commit is 1, the associated configuration may not be changed until reset.

Figure 7-37. MMR_CONFIG_LOCK Register

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

Table 7-37. MMR_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	globel_en_lock	R/W	0h	Locks the Global Enable Register.
7-6	RESERVED	R/W	Xh	
5	global_soft_rst_lock	R/W	0h	Locks the Global Soft Reset Register.
4-3	RESERVED	R/W	Xh	
2	errpin_lock	R/W	0h	Locks the Error Pin configuration registers. These are: Error Pin Control Register, Error Pin Counter Pre-Load Register, Error Pin PWM High Counter Pre-Load Register, Error Pin PWM Low Counter Pre-Load Register
1	errpin_mon_lock	R/W	0h	Locks the Error Pin Monitor Config Register.
0	hi_pri_wd_lock	R/W	0h	Locks the High Priority Watchdog configuration registers. These are: High Priority Watchdog Config Register, High Priority Watchdog Pre-Load Register.

7.7.2.20 MMR_CONFIG_COMMIT Register (Offset = 124h) [Reset = 000000XXh]

MMR_CONFIG_COMMIT is shown in Figure 7-38 and described in Table 7-38.

Return to the Summary Table.

The fields of the MMR Config Commit Register commit the lock configuration for the associated MMR Config Lock MMR bits. This prevents the Locks in the MMR Config Lock Register from changing. Once Commit is 1, the associated Lock and Commit may not be changed until reset.

Figure 7-38. MMR_CONFIG_COMMIT Register

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

Table 7-38. MMR_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	globel_en_commit	R/W	0h	Commits the lock for the Global Enable Register.
7-6	RESERVED	R/W	Xh	
5	global_soft_rst_commit	R/W	0h	Commits the lock for the Global Soft Reset Register.
4-3	RESERVED	R/W	Xh	
2	errpin_commit	R/W	0h	Commits the lock for the Error Pin configuration registers. These are: Error Pin Control Register, Error Pin Counter Pre-Load Register, Error Pin PWM High Counter Pre-Load Register, Error Pin PWM Low Counter Pre-Load Register
1	errpin_mon_commit	R/W	0h	Commits the lock for the Error Pin Monitor Config Register.
0	hi_pri_wd_commit	R/W	0h	Commits the lock for the High Priority Watchdog configuration registers. These are: High Priority Watchdog Config Register, High Priority Watchdog Pre-Load Register.

7.7.2.21 RAW_j Register (Offset = 400h + formula) [Reset = 00000000h]

RAW_j is shown in Figure 7-39 and described in Table 7-39.

Return to the Summary Table.

Raw Status/Set Register for Group A Errors

Offset = 400h + (j * 20h); where j = 0h to 7h

Figure 7-39. RAW_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sts																															
R/W1S-0h																															

Table 7-39. RAW_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	sts	R/W1S	0h	This is the raw status/set for errors Group N. Each bit corresponds to event Q where Q=N*32+Bit, e.g. bit 0 is event N*32+0, bit 1 is N*32+1, etc. This field is only reset by a Power-On-Reset, not warm reset. A global soft reset will set this field to 0. Read 0 is Inactive. Read 1 is Active/Pending. Write 0 has no effect. Write 1 sets the Interrupt Raw Status.

7.7.2.22 STS_j Register (Offset = 404h + formula) [Reset = 00000000h]

STS_j is shown in Figure 7-40 and described in Table 7-40.

Return to the Summary Table.

Error Enable and Clear Register

Offset = 404h + (j * 20h); where j = 0h to 7h

Figure 7-40. STS_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
msk																															
R/W1C-0h																															

Table 7-40. STS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	msk	R/W1C	0h	This is the masked status/clear for errors in Group N. Each bit corresponds to event Q where Q=N*32+Bit, e.g. bit 0 is event N*32+0, bit 1 is N*32+1, etc. This field is only reset by a Power-On-Reset, not warm reset. A global soft reset will set this field to 0. Read 0 is Inactive/Disabled. Read 1 is Active/Pending and Enabled. Write 0 has no effect. Write 1 clears the Interrupt Raw Status.

7.7.2.23 INTR_EN_SET_j Register (Offset = 408h + formula) [Reset = 00000000h]

INTR_EN_SET_j is shown in Figure 7-41 and described in Table 7-41.

Return to the Summary Table.

Error Enable Set Register

Offset = 408h + (j * 20h); where j = 0h to 7h

Figure 7-41. INTR_EN_SET_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
msk																															
R/W1S-0h																															

Table 7-41. INTR_EN_SET_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	msk	R/W1S	0h	This is the mask enable set for errors in Group N. Each bit corresponds to event Q where Q=N*32+Bit, e.g. bit 0 is event N*32+0, bit 1 is N*32+1, etc. If the corresponding bit and the global_enable are set, then the interrupt is unmasked. This field is only reset by a Power-On-Reset, not warm reset. A global soft reset will set this field to 0. Read 0 is Disabled. Read 1 is Enabled. Write 0 has no effect. Write 1 sets the Interrupt Enable.

7.7.2.24 INTR_EN_CLR_j Register (Offset = 40Ch + formula) [Reset = 00000000h]

INTR_EN_CLR_j is shown in Figure 7-42 and described in Table 7-42.

Return to the Summary Table.

Error Interrupt Enabled Clear register

Offset = 40Ch + (j * 20h); where j = 0h to 7h

Figure 7-42. INTR_EN_CLR_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
msk																															
R/W1C-0h																															

Table 7-42. INTR_EN_CLR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	msk	R/W1C	0h	This is the mask enable clear for errors in Group N. Each bit corresponds to event Q where Q=N*32+Bit, e.g. bit 0 is event N*32+0, bit 1 is N*32+1, etc. If the corresponding bit and the global_enable are set, then the interrupt is unmasked. This field is only reset by a Power-On-Reset, not warm reset. A global soft reset will set this field to 0. Read 0 is Disabled. Read 1 is Enabled. Write 0 has no effect. Write 1 clears the Interrupt Enable.

7.7.2.25 INT_PRIO_j Register (Offset = 410h + formula) [Reset = 00000000h]

INT_PRIO_j is shown in Figure 7-43 and described in Table 7-43.

Return to the Summary Table.

Error Interrupt Priority register

Offset = 410h + (j * 20h); where j = 0h to 7h

Figure 7-43. INT_PRIO_j Register

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

Table 7-43. INT_PRIO_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	msk	R/W	0h	This field is used to indicate which interrupt the corresonding event influences (if enabled) for event Group N. Each bit corresponds to event Q where Q=N*32+Bit, e.g. bit 0 is event N*32+0, bit 1 is N*32+1, etc. This field is only reset by a Power-On-Reset, not warm reset. A global soft reset will set this field to 0. A value of 0 means the event will affect the low priority interrupt, 1 the high priority interrupt.

7.7.2.26 CRIT_EN_SET_j Register (Offset = 800h + formula) [Reset = 00000000h]

CRIT_EN_SET_j is shown in Figure 7-44 and described in Table 7-44.

Return to the Summary Table.

Critical Priority Interrupt Enabled Set register

Offset = 800h + (j * 20h); where j = 0h to 7h

Figure 7-44. CRIT_EN_SET_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
msk																															
R/W1S-0h																															

Table 7-44. CRIT_EN_SET_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	msk	R/W1S	0h	This is the critical priority interrupt influence enable set for errors in Group N. Each bit corresponds to event Q where Q=N*32+Bit, e.g. bit 0 is event N*32+0, bit 1 is N*32+1, etc. If the corresponding bit and the global_enable are set, then the interrupt is unmasked. This field is only reset by a Power-On-Reset, not warm reset. A global soft reset will set this field to 0. Read 0 is Disabled. Read 1 is Enabled. Write 0 has no effect. Write 1 sets the Enable. The corresponding event, when set, will count as a pending event towards generating a critical priority interrupt.

7.7.2.27 CRIT_EN_CLR_j Register (Offset = 804h + formula) [Reset = 00000000h]

CRIT_EN_CLR_j is shown in Figure 7-45 and described in Table 7-45.

Return to the Summary Table.

Critical Priority Interrupt Enabled Clear register

Offset = 804h + (j * 20h); where j = 0h to 7h

Figure 7-45. CRIT_EN_CLR_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
msk																															
R/W1C-0h																															

Table 7-45. CRIT_EN_CLR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	msk	R/W1C	0h	This is the critical priority interrupt influence enable clear for errors in Group N. Each bit corresponds to event Q where Q=N*32+Bit, e.g. bit 0 is event N*32+0, bit 1 is N*32+1, etc. If the corresponding bit and the global_enable are set, then the interrupt is unmasked. This field is only reset by a Power-On-Reset, not warm reset. A global soft reset will set this field to 0. Read 0 is Disabled. Read 1 is Enabled. Write 0 has no effect. Write 1 clears the Enable. The corresponding event will no longer count as a pending event towards generating a critical priority interrupt.

7.7.3 ESM_SYSTEM_REGS Registers

Table 7-46 lists the memory-mapped registers for the ESM_SYSTEM_REGS registers. All register offset addresses not listed in Table 7-46 should be considered as reserved locations and the register contents should not be modified.

Table 7-46. ESM_SYSTEM_REGS Registers

Offset	Acronym	Register Name
0h	PID	Revision Register
4h	INFO	Info Register
8h	EN	Global Enable Register
Ch	SFT_RST	Global Soft Reset Register
20h	LOW_PRI	Low Priority Prioritized Register
28h	LOW	Low Priority Interrupt Status Register
30h	EOI	EOI Interrupt Register
40h	PIN_CTRL	Error Pin Control Register
44h	PIN_STS	Error Pin Status Register
48h	PIN_CNTR	Error Pin Counter Value Register
4Ch	PIN_CNTR_PRE	Error Pin Counter Value Pre-Load Register
50h	PWMH_PIN_CNTR	Error Pin PWM High Counter Value Register
54h	PWMH_PIN_CNTR_PRE	Error Pin PWM High Counter Value Pre-Load Register
58h	PWML_PIN_CNTR	Error Pin PWM Low Counter Value Register
5Ch	PWML_PIN_CNTR_PRE	Error Pin PWM Low Counter Value Pre-Load Register
A0h	ERRPIN_MON_CFG	Error Pin Monitor Config Register
A4h	ERRPIN_MON_INTR_SET	Error Pin Monitor Interrupt Status/Set Register
A8h	ERRPIN_MON_INTR_CLR	Error Pin Monitor Interrupt Status/Clear Register
100h	GROUP_N_LOCK	Group N Interrupt Lock Register
104h	GROUP_N_COMMIT	Group N Interrupt Commit Register
110h	ERR_PIN_INFLUENCE_LOCK	Error Pin Influence Lock Register
114h	ERR_PIN_INFLUENCE_COMMIT	Error Pin Influence Commit Register
118h	CRI_PRI_INFLUENCE_LOCK	Critical Prioirty Interrupt Influence Lock Register
11Ch	CRI_PRI_INFLUENCE_COMMIT	Critical Priority Interrupt Influence Lock Register
120h	MMR_CONFIG_LOCK	MMR Config Lock Register
124h	MMR_CONFIG_COMMIT	MMR Config Commit Register
400h + formula	RAW_j	Error Group N Event Raw Status/Set Register
404h + formula	STS_j	Error Group N Interrupt Enable Status/Clear Register
408h + formula	INTR_EN_SET_j	Error Group N Interrupt Enable Set Register
40Ch + formula	INTR_EN_CLR_j	Error Group N Interrupt Enabled Clear Register
410h + formula	INT_PRIO_j	Error Group N Interrupt Priority Register
414h + formula	PIN_EN_SET_j	Error Group N Error Pin Influence Set Register
418h + formula	PIN_EN_CLR_j	Error Group N Error Pin Influence Clear Register
800h + formula	CRIT_EN_SET_j	Error Group N Critical Priority Interrupt Influence Set Register
804h + formula	CRIT_EN_CLR_j	Error Group N Critical Priority Interrupt Influence Clear Register

Complex bit access types are encoded to fit into small table cells. Table 7-47 shows the codes that are used for access types in this section.

Table 7-47. ESM_SYSTEM_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

7.7.3.1 PID Register (Offset = 0h) [Reset = 6FE03101h]

PID is shown in Figure 7-46 and described in Table 7-48.

Return to the Summary Table.

The Revision Register contains the major and minor revisions for the module.

Figure 7-46. PID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
scheme		bu		func											
R-1h		R-2h		R-FE0h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rtl					major			custom		minor					
R-6h					R-1h			R-0h		R-1h					

Table 7-48. PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	scheme	R	1h	PID register scheme
29-28	bu	R	2h	Business Unit: 10 = Processors
27-16	func	R	FE0h	Module ID
15-11	rtl	R	6h	RTL revision. Will vary depending on release.
10-8	major	R	1h	Major revision
7-6	custom	R	0h	Custom
5-0	minor	R	1h	Minor revision

7.7.3.2 INFO Register (Offset = 4h) [Reset = XXXX0808h]

INFO is shown in Figure 7-47 and described in Table 7-49.

Return to the Summary Table.

The Info Register gives the configuration Inforrmation of this ESM.

Figure 7-47. INFO Register

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

Table 7-49. INFO Register Field Descriptions

Bit	Field	Type	Reset	Description
31	last_reset	R	0h	Indicates the Source of the last Reset 0 – Last reset was a Power On Reset 1 – Last reset was a Warm Reset
30	crit_intr	R	0h	Indicates if the critical priority interrupt output has asserted 0 – Critical Priority Interrupt output has not triggered 1 – Critical Priority Interrupt output has triggered
29-16	RESERVED	R	XXXh	
15-8	pulse_groups	R	8h	Number of Pulse Error Groups
7-0	groups	R	8h	Total number of Error Groups

7.7.3.3 EN Register (Offset = 8h) [Reset = 00000000h]

EN is shown in Figure 7-48 and described in Table 7-50.

Return to the Summary Table.

The Global Enable Register has the master interrupt mask

Figure 7-48. EN Register

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

Table 7-50. EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	key	R/W	0h	Global Enable 4'b0000- All interrupts are disabled. 4'b1111- All interrupts are enabled

7.7.3.4 SFT_RST Register (Offset = Ch) [Reset = 00000000h]

SFT_RST is shown in Figure 7-49 and described in Table 7-51.

Return to the Summary Table.

The Global Soft Reset Register controls the global clear for raw status and enables

Figure 7-49. SFT_RST Register

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

Table 7-51. SFT_RST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3-0	key	W	0h	Global Soft Reset 4'b1111- Clear all raw status and enable bits. All Others - No effect

7.7.3.5 LOW_PRI Register (Offset = 20h) [Reset = FFFFFFFFh]

LOW_PRI is shown in Figure 7-50 and described in Table 7-52.

Return to the Summary Table.

Shows which is the highest priority outstanding low priority interrupt

Figure 7-50. LOW_PRI Register

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

Table 7-52. LOW_PRI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	pls	R	FFFFh	This is the highest priority outstanding low priority pulse interrupt The lowest event number has the highest priority. A value of all ones (0xFFFF) indicates that there are no low priority interrupts pending. This field is updated whenever a new, higher priority event occurs
15-0	lvl	R	FFFFh	This is the highest priority outstanding low priority level interrupt The lowest event number has the highest priority. A value of all ones (0xFFFF) indicates that there are no low priority interrupts pending. This field is updated whenever a new, higher priority event occurs

7.7.3.6 LOW Register (Offset = 28h) [Reset = 00000000h]

LOW is shown in Figure 7-51 and described in Table 7-53.

Return to the Summary Table.

Shows which groups have oustanding low priority interrupts

Figure 7-51. LOW Register

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

Table 7-53. LOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	sts	R	0h	This is the raw status for low priority interrupt event groups Indicates which Event Groups have one or more Low Priority interrupts pending. This register is bit oriented where bit 0 is for Event Group 0, bit 1 is for Event Group 1 and bit N is for Event Group N

7.7.3.7 EOI Register (Offset = 30h) [Reset = 00000000h]

EOI is shown in Figure 7-52 and described in Table 7-54.

Return to the Summary Table.

End of Interrupt Register

Figure 7-52. EOI Register

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

Table 7-54. EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	W	0h	
10-0	key	W	0h	This is the interrupt being serviced 1h = Low Priority Error Interrupt 2h = High Priority Error Interrupt 5h = Error Pin Monitor Interrupt

7.7.3.8 PIN_CTRL Register (Offset = 40h) [Reset = 00000000h]

PIN_CTRL is shown in Figure 7-53 and described in Table 7-55.

Return to the Summary Table.

This register controls the error_pin_n output

Figure 7-53. PIN_CTRL Register

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

Table 7-55. PIN_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-8	polarity	R/W	0h	Level mode polarity. When the error output pin is a level, by default it is active low, driving the error pin low to indicate an error. When 4'F is written to this field, the polarity will be active high instead. Software should not write values other than 4'h0 and 4'hF Any other values will default to active low behavior.
7-4	pwm_en	R/W	0h	PWM enable 0h = Error output pin is a level Fh = Error output pin is a PWM signal
3-0	key	R/W	0h	Pin Control Key 0h = Normal Mode. Error Pin will activate when an enabled Error Event occurs 5h = Clear Event. Generates a Clear event to the ESM state machine. Will return to Normal Mode on the next cycle. Ah = Force Error Mode. Forces the Error Pin active. To clear, write this field back to Normal Mode. You may only write to Force Error Mode while in IDLE. Attempting Force Error while in another state will have no effect.

7.7.3.9 PIN_STS Register (Offset = 44h) [Reset = 00000000h]

PIN_STS is shown in Figure 7-54 and described in Table 7-56.

Return to the Summary Table.

This register reflects the status of the error_pin_n output

Figure 7-54. PIN_STS Register

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

Table 7-56. PIN_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	val	R	0h	Value of the error_pin_n In Error Pin Level mode: 1 – De-Asserted (pin is inactive, high) 0 – Asserted (pin is active, low)

7.7.3.10 PIN_CNTR Register (Offset = 48h) [Reset = 00FFFFFFh]

PIN_CNTR is shown in Figure 7-55 and described in Table 7-57.

Return to the Summary Table.

This register shows the current value of the error pin counter

Figure 7-55. PIN_CNTR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								count																							
R-0h								R-00FFFFFFh																							

Table 7-57. PIN_CNTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	count	R	00FFFFFFh	Current Counter Value

7.7.3.11 PIN_CNTR_PRE Register (Offset = 4Ch) [Reset = 00FFFFFFh]

PIN_CNTR_PRE is shown in Figure 7-56 and described in Table 7-58.

Return to the Summary Table.

This register contains the value that is loaded in to the Error Counter

Figure 7-56. PIN_CNTR_PRE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								count																							
R/W-0h								R/W-00FFFFFFh																							

Table 7-58. PIN_CNTR_PRE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-0	count	R/W	00FFFFFFh	Counter Pre-Load Value

7.7.3.12 PWMH_PIN_CNTR Register (Offset = 50h) [Reset = 00FFFFFFh]

PWMH_PIN_CNTR is shown in Figure 7-57 and described in Table 7-59.

Return to the Summary Table.

This register shows the current value of the error pin PWM high counter

Figure 7-57. PWMH_PIN_CNTR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								count																							
R-0h								R-00FFFFFFh																							

Table 7-59. PWMH_PIN_CNTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	count	R	00FFFFFFh	Current Counter Value

7.7.3.13 PWMH_PIN_CNTR_PRE Register (Offset = 54h) [Reset = 00FFFFFFh]

PWMH_PIN_CNTR_PRE is shown in Figure 7-58 and described in Table 7-60.

Return to the Summary Table.

This register contains the value that is loaded in to the Error PWM High Counter

Figure 7-58. PWMH_PIN_CNTR_PRE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								count																							
R/W-0h								R/W-00FFFFFFh																							

Table 7-60. PWMH_PIN_CNTR_PRE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-0	count	R/W	00FFFFFFh	Counter Pre-Load Value

7.7.3.14 PWML_PIN_CNTR Register (Offset = 58h) [Reset = 00FFFFFFh]

PWML_PIN_CNTR is shown in Figure 7-59 and described in Table 7-61.

Return to the Summary Table.

This register shows the current value of the error pin PWM low counter

Figure 7-59. PWML_PIN_CNTR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								count																							
R-0h								R-00FFFFFFh																							

Table 7-61. PWML_PIN_CNTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	count	R	00FFFFFFh	Current Counter Value

7.7.3.15 PWML_PIN_CNTR_PRE Register (Offset = 5Ch) [Reset = 00FFFFFFh]

PWML_PIN_CNTR_PRE is shown in Figure 7-60 and described in Table 7-62.

Return to the Summary Table.

This register contains the value that is loaded in to the Error PWM Low Counter

Figure 7-60. PWML_PIN_CNTR_PRE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								count																							
R/W-0h								R/W-00FFFFFFh																							

Table 7-62. PWML_PIN_CNTR_PRE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-0	count	R/W	00FFFFFFh	Counter Pre-Load Value

7.7.3.16 ERRPIN_MON_CFG Register (Offset = A0h) [Reset = 00000000h]

ERRPIN_MON_CFG is shown in Figure 7-61 and described in Table 7-63.

Return to the Summary Table.

The Error Pin Monitor Config Register is used to enable or disable the Error Pin Monitor and its associated interrupt.

Figure 7-61. ERRPIN_MON_CFG Register

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

Table 7-63. ERRPIN_MON_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	mon_en	R/W	0h	Enable field for the Error Pin Monitor. 0xA Enables the monitor. All other values disable the monitor

7.7.3.17 ERRPIN_MON_INTR_SET Register (Offset = A4h) [Reset = 00000000h]

ERRPIN_MON_INTR_SET is shown in Figure 7-62 and described in Table 7-64.

Return to the Summary Table.

The Error Pin Monitor Interrupt Status/Set Register indicates the status of the Error Pin Monitor Interrupt. This register is reset by the Power On Reset. A write to this register may allow software to set the interrupt output.

Figure 7-62. ERRPIN_MON_INTR_SET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															val
R/W-0h															R/W1S-0h

Table 7-64. ERRPIN_MON_INTR_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	val	R/W1S	0h	Read the current interrupt status. Write 1 to set the interrupt output

7.7.3.18 ERRPIN_MON_INTR_CLR Register (Offset = A8h) [Reset = 00000000h]

ERRPIN_MON_INTR_CLR is shown in Figure 7-63 and described in Table 7-65.

Return to the Summary Table.

The Error Pin Monitor Status/Clear Register indicates the status of the Error Pin Monitor Interrupt. This register is reset by the Power On Reset. A write to this register may allow software to temporarily clear the interrupt.

Figure 7-63. ERRPIN_MON_INTR_CLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															val
R/W-0h															R/W1C-0h

Table 7-65. ERRPIN_MON_INTR_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	val	R/W1C	0h	Read the current interrupt status. Write 1 to clear the interrupt output. The interrupt will reassert if the conditions that triggered it persist

7.7.3.19 GROUP_N_LOCK Register (Offset = 100h) [Reset = 00000000h]

GROUP_N_LOCK is shown in Figure 7-64 and described in Table 7-66.

Return to the Summary Table.

The fields of the Group N Interrupt Lock Register lock the configuration for the associated Error Group N Interrupt Enable Set/Clear Registers and the Error Group N Interrupt Priority Register. This locks the associated enable and priority bits for each Group for Low and High Priority Interrupts. Locks may be written until the associated Commit MMR fields are set to 1. Once Lock is 1 and Commit is 1, the associated configuration may not be changed until reset.

Figure 7-64. GROUP_N_LOCK Register

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

Table 7-66. GROUP_N_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	lock	R/W	0h	Each bit lock[N] bit will lock the associated masking MMRs associated with Group N. These are: Error Group N Interrupt Enabled Set Register, Error Group N Interrupt Enabled Clear Register, Error Group N Interrupt Priority Register

7.7.3.20 GROUP_N_COMMIT Register (Offset = 104h) [Reset = 00000000h]

GROUP_N_COMMIT is shown in Figure 7-65 and described in Table 7-67.

Return to the Summary Table.

The fields of the Group N Interrupt Commit Register commit the lock configuration for the associated Group N Interrupt Lock Register. This prevents the Locks in the Group N Interrupt Lock Register from changing. Once Commit is 1, the associated Lock and Commit may not be changed until reset.

Figure 7-65. GROUP_N_COMMIT Register

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

Table 7-67. GROUP_N_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	commit	R/W	0h	Each bit commit[N] bit will commit the lock configuration for the corresponding bit in the Group N Interrupt Lock Register.

7.7.3.21 ERR_PIN_INFLUENCE_LOCK Register (Offset = 110h) [Reset = 00000000h]

ERR_PIN_INFLUENCE_LOCK is shown in Figure 7-66 and described in Table 7-68.

Return to the Summary Table.

The fields of the Error Pin Influence Lock Register lock the Error Pin Influence configuration for the associated Error Groups. Locks may be written until the associated Commit MMR fields are set to 1. Once Lock is 1 and Commit is 1, the associated configuration may not be changed until reset.

Figure 7-66. ERR_PIN_INFLUENCE_LOCK Register

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

Table 7-68. ERR_PIN_INFLUENCE_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	lock	R/W	0h	Each bit lock[N] bit will lock the associated masking MMRs associated with Group N. These are: Error Group N Error Pin Influence Set Register, Error Group N Error Pin Influence Clear Register

7.7.3.22 ERR_PIN_INFLUENCE_COMMIT Register (Offset = 114h) [Reset = 00000000h]

ERR_PIN_INFLUENCE_COMMIT is shown in Figure 7-67 and described in Table 7-69.

Return to the Summary Table.

The fields of the Error Pin Influence Commit Register commit the lock configuration for the associated Error Groups. This prevents the Locks in the Error Pin Influence Lock Register from changing. Once Commit is 1, the associated Lock and Commit may not be changed until reset.

Figure 7-67. ERR_PIN_INFLUENCE_COMMIT Register

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

Table 7-69. ERR_PIN_INFLUENCE_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	commit	R/W	0h	Each bit commit[N] bit will commit the lock configuration for the corresponding bit in the Error Pin Influence Lock Register.

7.7.3.23 CRI_PRI_INFLUENCE_LOCK Register (Offset = 118h) [Reset = 00000000h]

CRI_PRI_INFLUENCE_LOCK is shown in Figure 7-68 and described in Table 7-70.

Return to the Summary Table.

The fields of the Critical Priority Interrupt Influence Lock Register lock the Critical Priority Interrupt Influence configuration for the associated Error Groups. Locks may be written until the associated Commit MMR fields are set to 1. Once Lock is 1 and Commit is 1, the associated configuration may not be changed until reset.

Figure 7-68. CRI_PRI_INFLUENCE_LOCK Register

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

Table 7-70. CRI_PRI_INFLUENCE_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	lock	R/W	0h	Each bit lock[N] bit will lock the associated masking MMRs associated with Group N. These are: Error Group N Critical Priority Interrupt Influence Set Register, Error Group N Critical Priority Interrupt Influence Clear Register

7.7.3.24 CRI_PRI_INFLUENCE_COMMIT Register (Offset = 11Ch) [Reset = 00000000h]

CRI_PRI_INFLUENCE_COMMIT is shown in Figure 7-69 and described in Table 7-71.

Return to the Summary Table.

The fields of the Critical Priority Interrupt Influence Commit Register commit the lock configuration for the associated Error Groups. This prevents the Locks in the Critical Priority Interrupt Influence Lock Register from changing. Once Commit is 1, the associated Lock and Commit may not be changed until reset.

Figure 7-69. CRI_PRI_INFLUENCE_COMMIT Register

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

Table 7-71. CRI_PRI_INFLUENCE_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	commit	R/W	0h	Each bit commit[N] bit will commit the lock configuration for the corresponding bit in the Critical Priority Interrupt Influence Lock Register.

7.7.3.25 MMR_CONFIG_LOCK Register (Offset = 120h) [Reset = 000000XXh]

MMR_CONFIG_LOCK is shown in Figure 7-70 and described in Table 7-72.

Return to the Summary Table.

The fields of the MMR Config Lock Register lock the configuration for the associated MMRs. This prevents changes once the lock configuration is committed with the MMR Config Commit Register. Locks may be written until the associated Commit MMR fields are set to 1. Once Lock is 1 and Commit is 1, the associated configuration may not be changed until reset.

Figure 7-70. MMR_CONFIG_LOCK Register

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

Table 7-72. MMR_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	globel_en_lock	R/W	0h	Locks the Global Enable Register.
7-6	RESERVED	R/W	Xh	
5	global_soft_rst_lock	R/W	0h	Locks the Global Soft Reset Register.
4-3	RESERVED	R/W	Xh	
2	errpin_lock	R/W	0h	Locks the Error Pin configuration registers. These are: Error Pin Control Register, Error Pin Counter Pre-Load Register, Error Pin PWM High Counter Pre-Load Register, Error Pin PWM Low Counter Pre-Load Register
1	errpin_mon_lock	R/W	0h	Locks the Error Pin Monitor Config Register.
0	hi_pri_wd_lock	R/W	0h	Locks the High Priority Watchdog configuration registers. These are: High Priority Watchdog Config Register, High Priority Watchdog Pre-Load Register.

7.7.3.26 MMR_CONFIG_COMMIT Register (Offset = 124h) [Reset = 000000XXh]

MMR_CONFIG_COMMIT is shown in Figure 7-71 and described in Table 7-73.

Return to the Summary Table.

The fields of the MMR Config Commit Register commit the lock configuration for the associated MMR Config Lock MMR bits. This prevents the Locks in the MMR Config Lock Register from changing. Once Commit is 1, the associated Lock and Commit may not be changed until reset.

Figure 7-71. MMR_CONFIG_COMMIT Register

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

Table 7-73. MMR_CONFIG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	globel_en_commit	R/W	0h	Commits the lock for the Global Enable Register.
7-6	RESERVED	R/W	Xh	
5	global_soft_rst_commit	R/W	0h	Commits the lock for the Global Soft Reset Register.
4-3	RESERVED	R/W	Xh	
2	errpin_commit	R/W	0h	Commits the lock for the Error Pin configuration registers. These are: Error Pin Control Register, Error Pin Counter Pre-Load Register, Error Pin PWM High Counter Pre-Load Register, Error Pin PWM Low Counter Pre-Load Register
1	errpin_mon_commit	R/W	0h	Commits the lock for the Error Pin Monitor Config Register.
0	hi_pri_wd_commit	R/W	0h	Commits the lock for the High Priority Watchdog configuration registers. These are: High Priority Watchdog Config Register, High Priority Watchdog Pre-Load Register.

7.7.3.27 RAW_j Register (Offset = 400h + formula) [Reset = 00000000h]

RAW_j is shown in Figure 7-72 and described in Table 7-74.

Return to the Summary Table.

Raw Status/Set Register for Group A Errors

Offset = 400h + (j * 20h); where j = 0h to 7h

Figure 7-72. RAW_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sts																															
R/W1S-0h																															

Table 7-74. RAW_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	sts	R/W1S	0h	This is the raw status/set for errors Group N. Each bit corresponds to event Q where Q=N*32+Bit, e.g. bit 0 is event N*32+0, bit 1 is N*32+1, etc. This field is only reset by a Power-On-Reset, not warm reset. A global soft reset will set this field to 0. Read 0 is Inactive. Read 1 is Active/Pending. Write 0 has no effect. Write 1 sets the Interrupt Raw Status.

7.7.3.28 STS_j Register (Offset = 404h + formula) [Reset = 00000000h]

STS_j is shown in Figure 7-73 and described in Table 7-75.

Return to the Summary Table.

Error Enable and Clear Register

Offset = 404h + (j * 20h); where j = 0h to 7h

Figure 7-73. STS_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
msk																															
R/W1C-0h																															

Table 7-75. STS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	msk	R/W1C	0h	This is the masked status/clear for errors in Group N. Each bit corresponds to event Q where Q=N*32+Bit, e.g. bit 0 is event N*32+0, bit 1 is N*32+1, etc. This field is only reset by a Power-On-Reset, not warm reset. A global soft reset will set this field to 0. Read 0 is Inactive/Disabled. Read 1 is Active/Pending and Enabled. Write 0 has no effect. Write 1 clears the Interrupt Raw Status.

7.7.3.29 INTR_EN_SET_j Register (Offset = 408h + formula) [Reset = 00000000h]

INTR_EN_SET_j is shown in Figure 7-74 and described in Table 7-76.

Return to the Summary Table.

Error Enable Set Register

Offset = 408h + (j * 20h); where j = 0h to 7h

Figure 7-74. INTR_EN_SET_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
msk																															
R/W1S-0h																															

Table 7-76. INTR_EN_SET_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	msk	R/W1S	0h	This is the mask enable set for errors in Group N. Each bit corresponds to event Q where Q=N*32+Bit, e.g. bit 0 is event N*32+0, bit 1 is N*32+1, etc. If the corresponding bit and the global_enable are set, then the interrupt is unmasked. This field is only reset by a Power-On-Reset, not warm reset. A global soft reset will set this field to 0. Read 0 is Disabled. Read 1 is Enabled. Write 0 has no effect. Write 1 sets the Interrupt Enable.

7.7.3.30 INTR_EN_CLR_j Register (Offset = 40Ch + formula) [Reset = 00000000h]

INTR_EN_CLR_j is shown in Figure 7-75 and described in Table 7-77.

Return to the Summary Table.

Error Interrupt Enabled Clear register

Offset = 40Ch + (j * 20h); where j = 0h to 7h

Figure 7-75. INTR_EN_CLR_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
msk																															
R/W1C-0h																															

Table 7-77. INTR_EN_CLR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	msk	R/W1C	0h	This is the mask enable clear for errors in Group N. Each bit corresponds to event Q where Q=N*32+Bit, e.g. bit 0 is event N*32+0, bit 1 is N*32+1, etc. If the corresponding bit and the global_enable are set, then the interrupt is unmasked. This field is only reset by a Power-On-Reset, not warm reset. A global soft reset will set this field to 0. Read 0 is Disabled. Read 1 is Enabled. Write 0 has no effect. Write 1 clears the Interrupt Enable.

7.7.3.31 INT_PRIO_j Register (Offset = 410h + formula) [Reset = 00000000h]

INT_PRIO_j is shown in Figure 7-76 and described in Table 7-78.

Return to the Summary Table.

Error Interrupt Priority register

Offset = 410h + (j * 20h); where j = 0h to 7h

Figure 7-76. INT_PRIO_j Register

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

Table 7-78. INT_PRIO_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	msk	R/W	0h	This field is used to indicate which interrupt the corresonding event influences (if enabled) for event Group N. Each bit corresponds to event Q where Q=N*32+Bit, e.g. bit 0 is event N*32+0, bit 1 is N*32+1, etc. This field is only reset by a Power-On-Reset, not warm reset. A global soft reset will set this field to 0. A value of 0 means the event will affect the low priority interrupt, 1 the high priority interrupt.

7.7.3.32 PIN_EN_SET_j Register (Offset = 414h + formula) [Reset = 00000000h]

PIN_EN_SET_j is shown in Figure 7-77 and described in Table 7-79.

Return to the Summary Table.

Error Pin Enabled Set register

Offset = 414h + (j * 20h); where j = 0h to 7h

Figure 7-77. PIN_EN_SET_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
msk																															
R/W1S-0h																															

Table 7-79. PIN_EN_SET_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	msk	R/W1S	0h	This is the error pin influence enable set for errors in Group N. Each bit corresponds to event Q where Q=N*32+Bit, e.g. bit 0 is event N*32+0, bit 1 is N*32+1, etc. If the corresponding bit and the global_enable are set, then the interrupt is unmasked. This field is only reset by a Power-On-Reset, not warm reset. A global soft reset will set this field to 0. Read 0 is Disabled. Read 1 is Enabled. Write 0 has no effect. Write 1 sets the Enable. The corresponding event, when set, will count as a pending error event for the ESM state machine.

7.7.3.33 PIN_EN_CLR_j Register (Offset = 418h + formula) [Reset = 00000000h]

PIN_EN_CLR_j is shown in Figure 7-78 and described in Table 7-80.

Return to the Summary Table.

Error Pin Enabled Clear register

Offset = 418h + (j * 20h); where j = 0h to 7h

Figure 7-78. PIN_EN_CLR_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
msk																															
R/W1C-0h																															

Table 7-80. PIN_EN_CLR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	msk	R/W1C	0h	This is the error pin influence enable clear for errors in Group N. Each bit corresponds to event Q where Q=N*32+Bit, e.g. bit 0 is event N*32+0, bit 1 is N*32+1, etc. If the corresponding bit and the global_enable are set, then the interrupt is unmasked. This field is only reset by a Power-On-Reset, not warm reset. A global soft reset will set this field to 0. Read 0 is Disabled. Read 1 is Enabled. Write 0 has no effect. Write 1 clears the Enable. The corresponding event will no longer count as a pending error event for the ESM state machine.

7.7.3.34 CRIT_EN_SET_j Register (Offset = 800h + formula) [Reset = 00000000h]

CRIT_EN_SET_j is shown in Figure 7-79 and described in Table 7-81.

Return to the Summary Table.

Critical Priority Interrupt Enabled Clear register

Offset = 800h + (j * 20h); where j = 0h to 7h

Figure 7-79. CRIT_EN_SET_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
msk																															
R/W1S-0h																															

Table 7-81. CRIT_EN_SET_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	msk	R/W1S	0h	This is the critical priority interrupt influence enable set for errors in Group N. Each bit corresponds to event Q where Q=N*32+Bit, e.g. bit 0 is event N*32+0, bit 1 is N*32+1, etc. If the corresponding bit and the global_enable are set, then the interrupt is unmasked. This field is only reset by a Power-On-Reset, not warm reset. A global soft reset will set this field to 0. Read 0 is Disabled. Read 1 is Enabled. Write 0 has no effect. Write 1 sets the Enable. The corresponding event, when set, will count as a pending event towards generating a critical priority interrupt.

7.7.3.35 CRIT_EN_CLR_j Register (Offset = 804h + formula) [Reset = 00000000h]

CRIT_EN_CLR_j is shown in Figure 7-80 and described in Table 7-82.

Return to the Summary Table.

Critical Priority Interrupt Enabled Clear register

Offset = 804h + (j * 20h); where j = 0h to 7h

Figure 7-80. CRIT_EN_CLR_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
msk																															
R/W1C-0h																															

Table 7-82. CRIT_EN_CLR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	msk	R/W1C	0h	This is the critical priority interrupt influence enable clear for errors in Group N. Each bit corresponds to event Q where Q=N*32+Bit, e.g. bit 0 is event N*32+0, bit 1 is N*32+1, etc. If the corresponding bit and the global_enable are set, then the interrupt is unmasked. This field is only reset by a Power-On-Reset, not warm reset. A global soft reset will set this field to 0. Read 0 is Disabled. Read 1 is Enabled. Write 0 has no effect. Write 1 clears the Enable. The corresponding event will no longer count as a pending event towards generating a critical priority interrupt.

7.7.4 ESM_SAFETYAGG_REGS Registers

Table 7-83 lists the memory-mapped registers for the ESM_SAFETYAGG_REGS registers. All register offset addresses not listed in Table 7-83 should be considered as reserved locations and the register contents should not be modified.

Table 7-83. ESM_SAFETYAGG_REGS Registers

Offset	Acronym	Register Name
0h	rev	Aggregator Revision Register
8h	vector	ECC Vector Register
Ch	stat	Misc Status
10h + formula	reserved_svbus_y	Reserved Area for Serial VBUS Registers
13Ch	ded_eoi_reg	EOI Register
140h	ded_status_reg0	Interrupt Status Register 0
180h	ded_enable_set_reg0	Interrupt Enable Set Register 0
1C0h	ded_enable_clr_reg0	Interrupt Enable Clear Register 0
200h	aggr_enable_set	AGGR interrupt enable set Register
204h	aggr_enable_clr	AGGR interrupt enable clear Register
208h	aggr_status_set	AGGR interrupt status set Register
20Ch	aggr_status_clr	AGGR interrupt status clear Register

Complex bit access types are encoded to fit into small table cells. Table 7-84 shows the codes that are used for access types in this section.

Table 7-84. ESM_SAFETYAGG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Wdecr	W decr	Write
Wincr	W incr	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

7.7.4.1 rev Register (Offset = 0h) [Reset = 66A03A01h]

rev is shown in Figure 7-81 and described in Table 7-85.

Return to the Summary Table.

Revision parameters

Figure 7-81. rev Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
scheme		bu		module_id											
R-1h		R-2h		R-6A0h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
revrtl					revmaj			custom		revmin					
R-7h					R-2h			R-0h		R-1h					

Table 7-85. rev Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	scheme	R	1h	Scheme
29-28	bu	R	2h	bu
27-16	module_id	R	6A0h	Module ID
15-11	revrtl	R	7h	RTL version
10-8	revmaj	R	2h	Major version
7-6	custom	R	0h	Custom version
5-0	revmin	R	1h	Minor version

7.7.4.2 vector Register (Offset = 8h) [Reset = 0000XX00h]

vector is shown in Figure 7-82 and described in Table 7-86.

Return to the Summary Table.

ECC Vector Register

Figure 7-82. vector Register

31	30	29	28	27	26	25	24
RESERVED							rd_svbus_done
R/W-0h							R/W1C-0h
23	22	21	20	19	18	17	16
rd_svbus_address							
R/W-0h							
15	14	13	12	11	10	9	8
rd_svbus	RESERVED				ecc_vector		
R/W1S-0h	R/W-Xh				R/W-0h		
7	6	5	4	3	2	1	0
ecc_vector							
R/W-0h							

Table 7-86. vector Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R/W	0h	
24	rd_svbus_done	R/W1C	0h	Status to indicate if read on serial VBUS is complete, write of any value will clear this bit.
23-16	rd_svbus_address	R/W	0h	Read address
15	rd_svbus	R/W1S	0h	Write 1 to trigger a read on the serial VBUS
14-11	RESERVED	R/W	Xh	
10-0	ecc_vector	R/W	0h	Value written to select the corresponding ESM Instance for control or status 0x0 - SYS ESM 0x1- ESM CPU1 0x2 - ESM CPU2 0x3 - ESM CPU3

7.7.4.3 stat Register (Offset = Ch) [Reset = 00000004h]

stat is shown in Figure 7-83 and described in Table 7-87.

Return to the Summary Table.

Misc Status

Figure 7-83. stat Register

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

Table 7-87. stat Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10-0	num_rams	R	4h	Indicates the number of ESM Instances serviced by the Safety Aggregator

7.7.4.4 reserved_svbus_y Register (Offset = 10h + formula) [Reset = 00000000h]

reserved_svbus_y is shown in Figure 7-84 and described in Table 7-88.

Return to the Summary Table.

Reference other documents that contain the ECC RAM wrapper and EDC controller serial vbus register sets.

Offset = 10h + (y * 4h); where y = 0h to 7h

Figure 7-84. reserved_svbus_y Register

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

Table 7-88. reserved_svbus_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	data	R/W	0h	This register displays the Serial VBUS register data

7.7.4.5 ded_eoi_reg Register (Offset = 13Ch) [Reset = 00000000h]

ded_eoi_reg is shown in Figure 7-85 and described in Table 7-89.

Return to the Summary Table.

EOI Register

Figure 7-85. ded_eoi_reg Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							eoi_wr
R/W-0h							R/W1S-0h

Table 7-89. ded_eoi_reg Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	eoi_wr	R/W1S	0h	EOI Register

7.7.4.6 ded_status_reg0 Register (Offset = 140h) [Reset = 00000000h]

ded_status_reg0 is shown in Figure 7-86 and described in Table 7-90.

Return to the Summary Table.

Interrupt Status Register 0

Figure 7-86. ded_status_reg0 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				esmss_esm2_edc_ctrl_busecc_pend	esmss_esm1_edc_ctrl_busecc_pend	esmss_esm0_edc_ctrl_busecc_pend	esmss_sys_esm_edc_ctrl_busecc_pend
R/W-0h				R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h

Table 7-90. ded_status_reg0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	esmss_esm2_edc_ctrl_busecc_pend	R/W1S	0h	Interrupt Pending Status for ESM CPU3 EDC Controller
2	esmss_esm1_edc_ctrl_busecc_pend	R/W1S	0h	Interrupt Pending Status for ESM CPU2 EDC Controller
1	esmss_esm0_edc_ctrl_busecc_pend	R/W1S	0h	Interrupt Pending Status for ESM CPU1 EDC Controller
0	esmss_sys_esm_edc_ctrl_busecc_pend	R/W1S	0h	Interrupt Pending Status for SYS ESM EDC Controller

7.7.4.7 ded_enable_set_reg0 Register (Offset = 180h) [Reset = 00000000h]

ded_enable_set_reg0 is shown in Figure 7-87 and described in Table 7-91.

Return to the Summary Table.

Interrupt Enable Set Register 0

Figure 7-87. ded_enable_set_reg0 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				esmss_esm2_edc_ctrl_busecc_enable_set	esmss_esm1_edc_ctrl_busecc_enable_set	esmss_esm0_edc_ctrl_busecc_enable_set	esmss_sys_esm_edc_ctrl_busecc_enable_set
R/W-0h				R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h

Table 7-91. ded_enable_set_reg0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	esmss_esm2_edc_ctrl_busecc_enable_set	R/W1S	0h	Interrupt Enable Set Register for ESM CPU3 EDC Controller
2	esmss_esm1_edc_ctrl_busecc_enable_set	R/W1S	0h	Interrupt Enable Set Register for ESM CPU2 EDC Controller
1	esmss_esm0_edc_ctrl_busecc_enable_set	R/W1S	0h	Interrupt Enable Set Register for ESM CPU1 EDC Controller
0	esmss_sys_esm_edc_ctrl_busecc_enable_set	R/W1S	0h	Interrupt Enable Set Register for SYS ESM EDC Controller

7.7.4.8 ded_enable_clr_reg0 Register (Offset = 1C0h) [Reset = 00000000h]

ded_enable_clr_reg0 is shown in Figure 7-88 and described in Table 7-92.

Return to the Summary Table.

Interrupt Enable Clear Register 0

Figure 7-88. ded_enable_clr_reg0 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				esmss_esm2_edc_ctrl_busecc_enable_clr	esmss_esm1_edc_ctrl_busecc_enable_clr	esmss_esm0_edc_ctrl_busecc_enable_clr	esmss_sys_esm_edc_ctrl_busecc_enable_clr
R/W-0h				R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

Table 7-92. ded_enable_clr_reg0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	esmss_esm2_edc_ctrl_busecc_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ESM CPU3 EDC Controller
2	esmss_esm1_edc_ctrl_busecc_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ESM CPU2 EDC Controller
1	esmss_esm0_edc_ctrl_busecc_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ESM CPU1 EDC Controller
0	esmss_sys_esm_edc_ctrl_busecc_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for SYS ESM EDC Controller

7.7.4.9 aggr_enable_set Register (Offset = 200h) [Reset = 00000000h]

aggr_enable_set is shown in Figure 7-89 and described in Table 7-93.

Return to the Summary Table.

AGGR interrupt enable set Register

Figure 7-89. aggr_enable_set Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						timeout	parity
R/W-0h						R/W1S-0h	R/W1S-0h

Table 7-93. aggr_enable_set Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	timeout	R/W1S	0h	interrupt enable set for svbus timeout errors
0	parity	R/W1S	0h	interrupt enable set for parity errors

7.7.4.10 aggr_enable_clr Register (Offset = 204h) [Reset = 00000000h]

aggr_enable_clr is shown in Figure 7-90 and described in Table 7-94.

Return to the Summary Table.

AGGR interrupt enable clear Register

Figure 7-90. aggr_enable_clr Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						timeout	parity
R/W-0h						R/W1C-0h	R/W1C-0h

Table 7-94. aggr_enable_clr Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	timeout	R/W1C	0h	interrupt enable clear for svbus timeout errors
0	parity	R/W1C	0h	interrupt enable clear for parity errors

7.7.4.11 aggr_status_set Register (Offset = 208h) [Reset = 00000000h]

aggr_status_set is shown in Figure 7-91 and described in Table 7-95.

Return to the Summary Table.

AGGR interrupt status set Register

Figure 7-91. aggr_status_set Register

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

Table 7-95. aggr_status_set Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-2	timeout	R/Wincr	0h	interrupt status set for svbus timeout errors 0 - No timeout errors have occurred 1 - 1 timeout has occurred 2 - 2 timeout have occurred 3 - 3 or more timeouts have occurred Write of a non-zero value increments that many from timeout error fields
1-0	parity	R/Wincr	0h	interrupt status set for parity errors 0 - No parity errors have occurred 1 - 1 parity error has occurred 2 - 2 parity errors have occurred 3 - 3 or more parity errors have occurred Write of a non-zero value increments that many from parity error fields

7.7.4.12 aggr_status_clr Register (Offset = 20Ch) [Reset = 00000000h]

aggr_status_clr is shown in Figure 7-92 and described in Table 7-96.

Return to the Summary Table.

AGGR interrupt status clear Register

Figure 7-92. aggr_status_clr Register

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

Table 7-96. aggr_status_clr Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-2	timeout	R/Wdecr	0h	interrupt status clear for svbus timeout errors 0 - No timeout errors have occurred 1 - 1 timeout has occurred 2 - 2 timeout have occurred 3 - 3 or more timeouts have occurred Write of a non-zero value decrements that many from timeout error fields
1-0	parity	R/Wdecr	0h	interrupt status clear for parity errors 0 - No parity errors have occurred 1 - 1 parity error has occurred 2 - 2 parity errors have occurred 3 - 3 or more parity errors have occurred Write of a non-zero value decrements that many from parity error fields

Chapter 8

Error Aggregator

This chapter describes the features and operation of the error aggregator module. The error aggregator is an interface module that aggregates and stores error information needed for critical errors in the device. The aggregated error output is passed on to ESM (Error Signalling Module) to take appropriate/configurable action on the error occurrence.

8.1 Introduction

Error aggregator is an interface module between peripherals that generate various errors like C29x, RTDMA, Memory controllers, Peripheral bridges, Read Interfaces, and ESM (Error Signaling Module). The main purpose of the Error Aggregator is to accumulate various errors from various sources and provide aggregated error output to ESM module. The module also logs the error status, error address, and error type information while segregating high-priority errors and low-priority errors for ESM to generate appropriate action to alert the CPU.

8.2 Error Aggregator Modules

Each Error Source provides the following information for all Error Aggregator Modules:

- Error - Pulse signal is generated on the occurrence of any error sent to ESM for further action.
- Error Address - System Address at which the error occurred used to detect and debug the error origin.
- Error Type - Multibit signal that indicates the type of error used to classify the error into predefined categories outlined later in the chapter.

All CPU Error Aggregator Modules additionally provide a Program Counter (PC) log for first high-priority error occurrence.

Figure 8-1 illustrates the module working and implementation. Each Aggregator module aggregates an error from various sources. Upon error occurrence, the corresponding error address and type are logged into the error address and error type registers, respectively.

The errors are classified as high or low priority based on the list in Section 8.6.

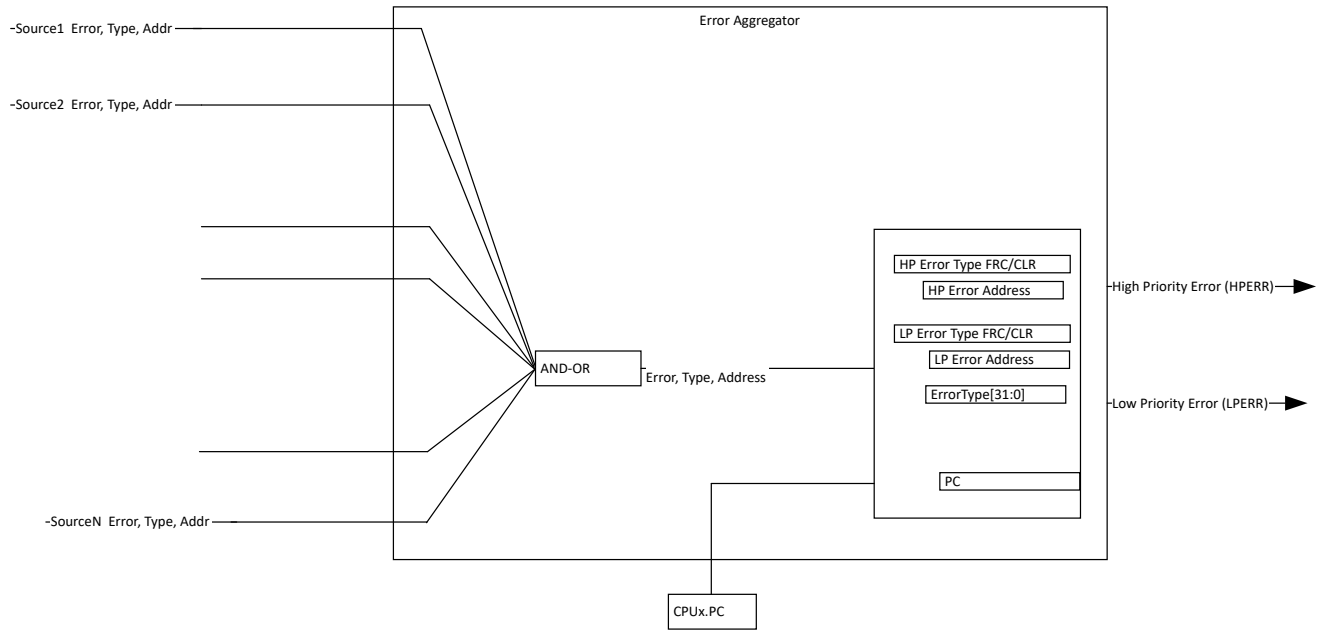


Figure 8-1. Error Aggregator Block Diagram

Error Aggregator modules implemented in the device are:

1. CPUx PR Error Aggregator - Aggregates errors occurred during CPUx program fetch access
2. CPUx DR1 Error Aggregator - Aggregates errors occurred during CPUx Data Read access on DR1 port
3. CPUx DR2 Error Aggregator - Aggregates errors occurred during CPUx Data Read access on DR2 port
4. CPUx DW Error Aggregator - Aggregates errors occurred during CPUx data write access
5. CPUx INT Error Aggregator - Aggregates interrupt related errors from CPUx and associated PIPE module
6. RTDMAx DR Error Aggregator - Aggregates errors occurred during RTDMAx data read access
7. RTDMAx DW Error Aggregator - Aggregates errors occurred during RTDMAx data write access
8. SSU Error Aggregator - Aggregates errors sent out by SSU module

9. EtherCAT Error Aggregator - Aggregates errors occurred during EtherCAT memory access
10. HSM Error Aggregator - Aggregates errors sent out by HSM subsystem

Note

x indicates that each error aggregator is repeated per initiator instance. EtherCAT only provides error and error address information so error type is tied off to uncorrectable error (0x40).

8.3 Error Propagation Path from Source to CPU

The Section 8.3 shown below describes the device level error handling and propagation from source to CPU. Error events from Error aggregator combined with other device level error events are provided to ESM (Error Signalling Module). Please refer to ESM error event map table for the complete list of error sources. ESM consolidates the responses to the error events and combined with other device level interrupt sources are sent to PIPE. For complete list of interrupts sent to PIPE refer to the PIPE channel mapping table. Optionally the error events can be configured to signal error pin output, trigger reset to device or respective CPU depending on the application through ESM. Refer to ESM chapter for more details.

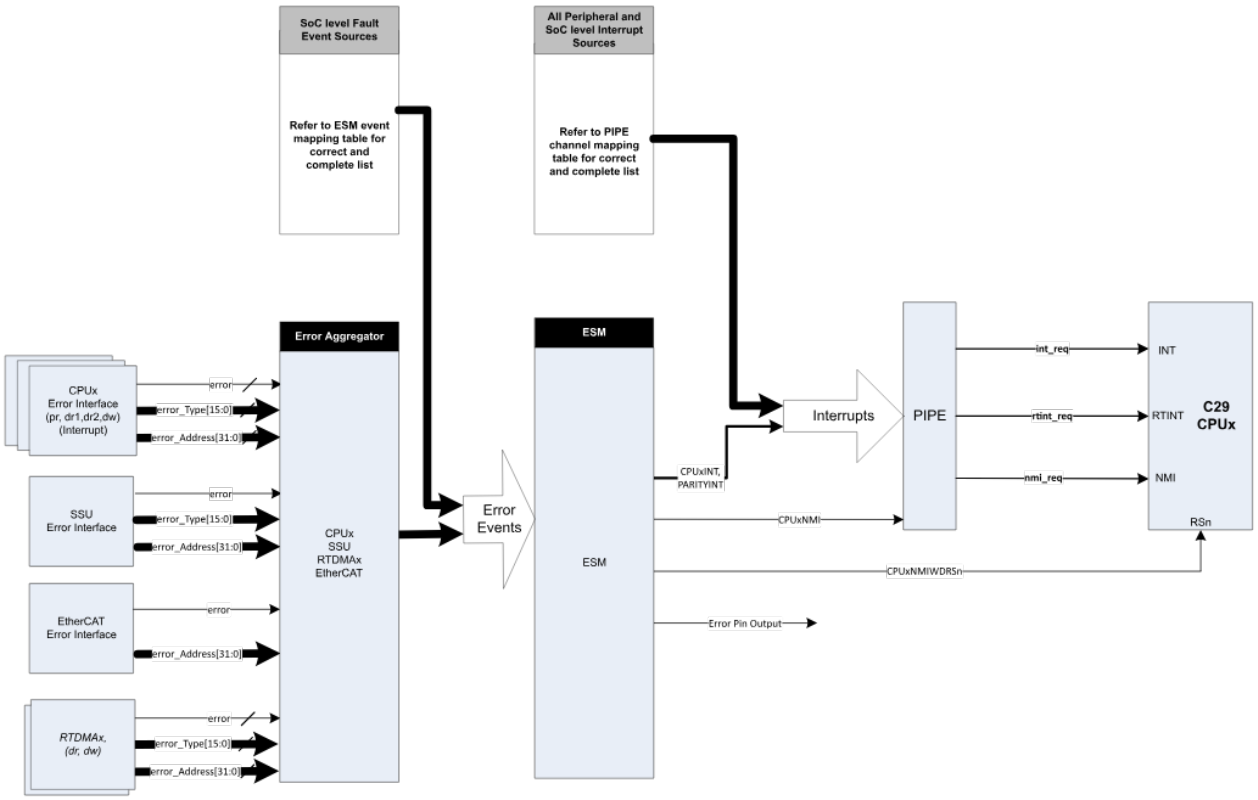


Figure 8-2. Error Propagation from Source to CPU

8.4 Error Aggregator Interface

This section provides details on how error information is handled and interfaced to the Error Signaling Module (ESM). The error outputs from multiple error aggregators are ORed and applied as single source to ESM. These includes:

- Low-Priority Errors from CPUx PR, CPUx DR1, CPUx DR2, CPUx DW are combined as CPUx LPERR

- High-Priority Errors from CPUx PR, CPUx DR1, CPUx DR2, CPUx DW are combined as CPUx HPERR
- Low-Priority Errors from RTDMAx DR + RTDMAx DW are combined as RTDMAx LPERR
- High-Priority Errors from RTDMAx DR + RTDMAx DW are combined as RTDMAx HPERR

Figure 8-3 shows a conceptual block diagram of the module functionality and how the output flag from each aggregator is combined as described in points above. The block diagram does not show all error aggregator modules available in the system, refer to the detailed list of supported error aggregators.

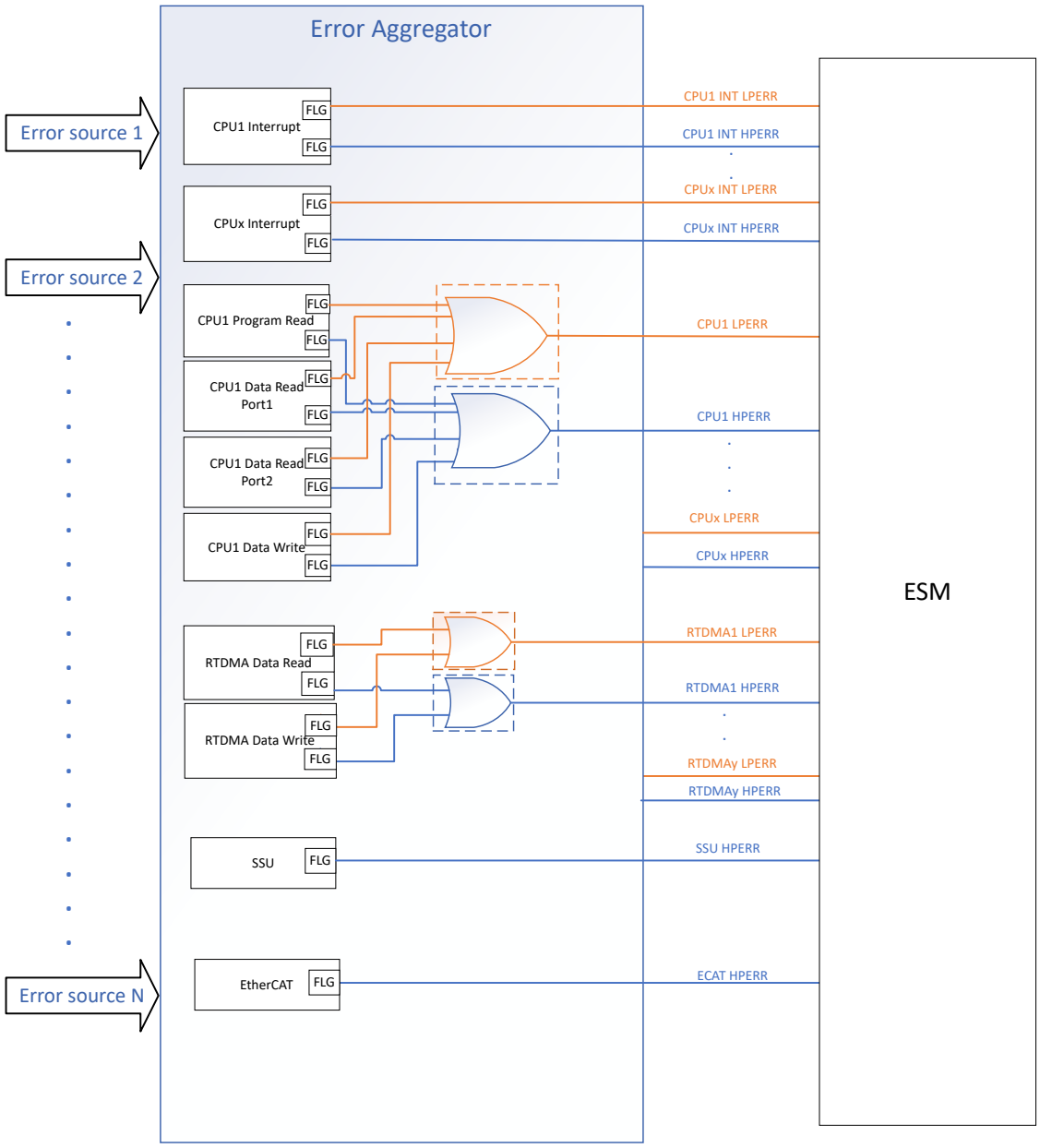


Figure 8-3. Error Aggregator Interface

8.4.1 Functional Description

The address at which the first error occurred and the corresponding error type are logged into the error address and error type registers, respectively. The errors are categorized into high- and low-priority errors.

Following actions are taken by the Error Aggregator on the occurrence of any high-priority error:

- Error type bit corresponding to the error occurred is set.
- High-Priority Error output pulse is generated.
- Address at which the first high-priority error occurred is latched to the High-Priority Error Address register. This register is not updated on further error occurrence until the high-priority errors in the error type register are cleared.
- On further occurrence of high-priority errors, the error type register continues to accumulate new errors but the error address register is not updated and no further high-priority error output pulses are generated until all the high-priority errors in the error type register are cleared.
- Additionally the user can also emulate or force an error condition by writing to the corresponding error_type in the ERROR_TYPE_FRC register.

Following actions are taken by the Error Aggregator on the occurrence of any low-priority error:

- Error type bit corresponding to the error occurred is set.
- Low-Priority Error output pulse is generated.
- Address at which the first low-priority error occurred is latched to the Low-Priority Error Address register. This register is not updated on further error occurrence until the low-priority errors in the error type register are cleared.
- On further occurrence of low-priority errors, the error type register continues to accumulate new errors but the error address register is not updated and no further low-priority error output pulses are generated until all the low-priority errors in the error type register are cleared.
- Additionally the user can also emulate or force an error condition by writing to the corresponding error_type in the ERROR_TYPE_FRC register.

  The error type register is cleared by writing to the ERROR_TYPE_CLR register.

  In addition to the above error aggregator logs, the program counter value at the time of occurrence of the first high-priority error is applicable to all CPUx (PR, DR1, DR2, DW, and INT) Error Aggregators only.

Note

The program counter value logged is not always a precise value. In some cases, the error address value is more precise than the program counter value.

8.5 Error Condition Handling User Guide

User's configure the NMI ISR which are output from ESM to do the following checks for errors passed on from Error Aggregator to the ESM:

1. Check the ESM Raw Status Register - Error Group N Event Raw Status/Set Register to check which events caused the error interrupt to trigger from ESM using the ESM Event Map.
2. If the error trigger is caused due to an error aggregator event mapped to ESM (Group 0 Events), then check the corresponding error aggregator Instance Type register.
   a. For example: If the Error Event 0 (ErrorAggregator_CPU1_HPERR) is detected from ESM in Step 1, user configures ISR to check error aggregator registers - CPU1_PR_ERROR_TYPE, CPU1_DR1_ERROR_TYPE, CPU1_DR2_ERROR_TYPE and CPU1_DW_ERROR_TYPE to determine what type of error was detected and in which instance of the error aggregator.
3. From Step 2, when user knows the error type, to narrow down the location of error user reads the corresponding error address register and program counter register.
   a. For example: If user finds out that the ILLEGAL_INSTRUCTION (Bit 10) in CPU1_PR_ERROR_TYPE is set, then user reads the corresponding CPU1_PR_HIGHPRIO_ERROR_ADDRESS register to find out the address at which the first high-priority error occurred on CPU1_PR access.

b. Also user reads CPU1_PR_PC register that contains the program counter value at the first high-priority error event.

4. While exiting the ISR, if user wants to clear the error when the source of the error is cleared - From Step 2, user stores the value of the error type register then copies the value to the error type clear register. If the source of the error still persists after the error clear, then the error aggregator generates another error pulse trigger to ESM.
   a. For example: From Step 3, user stores the value from the CPU1_PR_ERROR_TYPE register and copies the same value to the CPU1_PR_ERROR_TYPE_CLR register to clear the CPU1_PR_ERROR_TYPE register.

Note

If user does not do step 4, the error aggregator does not generate another error event input to ESM until corresponding priority error type register is cleared.

When the application is not able to clear the error before a NMIWD (High Priority Watchdog) timeout, then a reset is triggered from ESMCPU1 instance (refer to Figure 7-8). In this case, BootROM clears errors to avoid a back-to-back NMIWD rest loop and stores the error information and status to M0 RAM (refer to Table 4-35) for further debug.

In above case, BootROM clears the following status:
1. ESM Group0 RAW Status for ESMCPU1 and SYSESM instances of ESM-Subsystem
2. All CPUx error aggregator type registers

Also saves the following in M0 RAM for user to debug the source of error:
1. ESM RAW Status for Group0 only
2. Error Aggregator CPU1 - PR, DR1/2, DW, and INT instances error information including high-priority error address, low-priority error address, error type, and program counter registers

8.6 Error Type Information

This section lists the error type information and associated priority for all error aggregators.

Table 8-1. Errors on CPUx PR Interfaces

Error Type	CPUx PR Error	RAM, ROM, FRI – PR Error	Priority
0x01	Instruction fetch security violation. Instruction packet crossed LINK, STACK, ZONE boundary. Linear code crossed LINK, STACK, ZONE boundary. Regular Branch and Calls crossed STACK, ZONE boundary.	Reserved	High
0x02	Secure entry error	Reserved	High
0x04	Secure exit error	Reserved	High
0x08	MAX PSP error	Reserved	High
0x10	Access timeout error	Reserved	High
0x20	Access ACK error	Access ACK error	High
0x40	Uncorrectable error	Uncorrectable error	High
0x80	Correctable error	Reserved	Low
0x100	WARN PSP error	Reserved	Low
0x200	Software breakpoint error	Reserved	High
0x400	Illegal instruction error	Reserved	High
0x800	Instruction timeout error	Reserved	High

Table 8-2. Errors on DW Interface

Error Type	CPUx DW	RAM, Bridges DW	RTDMAx DW	Priority
0x01	Security violation	Reserved	Security violation	High
0x02	Reserved	Reserved	Reserved	-
0x04	Reserved	Reserved	Reserved	-
0x08	Reserved	Reserved	Reserved	-
0x10	Access timeout error	Reserved	Access timeout error	High
0x20	Access ACK error	Access ACK error	Access ACK error	High
0x40	Reserved	Uncorrectable error	Reserved	High
0x80	Reserved	Correctable error (RAM- RMW only)	Reserved	Low
0x100	Unaligned address error	Reserved	Reserved	High

Table 8-3. Errors on DR Interface

Error Type	CPUx DR1/DR2	RAM, Bridges, FRI - DR1/DR2/DR	RTDMAx DR	Priority
0x01	Security violation	Reserved	Security violation	High
0x02	Reserved	Reserved	Reserved	-
0x04	Reserved	Reserved	Reserved	-
0x08	Reserved	Reserved	Reserved	-
0x10	Access timeout error	Reserved	Access timeout error	High
0x20	Access ACK error	Access ACK error	Access ACK error	High
0x40	Uncorrectable error	Uncorrectable error	Uncorrectable error	High
0x80	Correctable error	Correctable error	Correctable error	Low
0x100	Unaligned address error	Reserved	Reserved	High

Table 8-4. Errors on CPUx INT Interface

Error Type	CPUx INT	Priority
0x01	MAIN ISR ENTRY ERROR	High
0x02	MAIN CORRECTABLE VECTOR ERROR	Low
0x04	MAIN UNCORRECTABLE VECTOR ERROR	High
0x08	MAIN INTERRUPT RETURN ERROR	High
0x10	INT ISR ENTRY ERROR	High
0x20	INT CORRECTABLE VECTOR ERROR	Low
0x40	INT UNCORRECTABLE VECTOR ERROR	High
0x80	INT INTERRUPT RETURN ERROR	High
0x100	RTINT ISR ENTRY ERROR	High
0x200	RTINT CORRECTABLE VECTOR ERROR	Low
0x400	RTINT UNCORRECTABLE VECTOR ERROR	High
0x800	RTINT INTERRUPT RETURN ERROR	High
0x1000	RTINT CONTEXT RESTORE CORRECTABLE VECTOR ERROR	Low
0x2000	RTINT CONTEXT RESTORE UNCORRECTABLE VECTOR ERROR	High
0x4000	Reserved	-
0x8000	NMI MAXISP ERROR	High
0x1 0000	NMI ISR ENTRY ERROR	High
0x2 0000	NMI CORRECTABLE VECTOR ERROR	Low
0x4 0000	NMI UNCORRECTABLE VECTOR ERROR	High

Table 8-4. Errors on CPUx INT Interface (continued)

Error Type	CPUx INT	Priority
0x8 0000	NMI INTERRUPT RETURN ERROR	High
0x10 0000	NMI CONTEXT RESTORE CORRECTABLE VECTOR ERROR	Low
0x20 0000	NMI CONTEXT RESTORE UNCORRECTABLE VECTOR ERROR	High
0x40 0000	VECTOR CORRECTABLE ERROR	Low
0x80 0000	VECTOR UNCORRECTABLE ERROR	High
0x100 0000	PIPE RTINT WARNISP	Low
0x200 0000	PIPE RTINT MAXISP	High
0x400 0000	PIPE SECURITY VIOLATION	High
0x800 0000	PIPE REGISTER PARITY ERROR	High
0x1000 0000	PIPE REGISTER PARITY DIAG ERROR	Low
0x2000 0000	PIPE LOCK_KEY ERROR	Low

Table 8-5. Errors on SSU

Error Type	SSU Error	Priority
0x01	Reserved	-
0x02	CPU1_SSU_MMR_ACCESS_ERROR	High
0x04	CPU2_SSU_MMR_ACCESS_ERROR	High
0x08	CPU3_SSU_MMR_ACCESS_ERROR	High
0x10	Reserved	-
0x20	Reserved	-
0x40	Reserved	-
0x80	BANKMAP/SECVALID/BANKMODE/SSUMODE invalid	High
0x100	FLC1_ILLADDR	High
0x300	FLC1_ILLPROG	High
0x500	FLC1_ILLERASE	High
0x700	FLC1_ILLRDVER	High
0x900	FLC1_ILLMODECH	High
0xB00	FLC1_ILLCMD	High
0xD00	FLC1_ILLSIZE	High
0xF00	FLC1_ILLBANKERASE	High
0x1000	FLC1_MMR_ACCESS_ERROR	High
0x2000	FLC2_ILLADDR	High
0x6000	FLC2_ILLPROG	High
0xA000	FLC2_ILLERASE	High
0xC000	FLC2_ILLRDVER	High
0x1 2000	FLC2_ILLMODECH	High
0x1 6000	FLC2_ILLCMD	High
0x1 A000	FLC2_ILLSIZE	High
0x1 C000	FLC2_ILLBANKERASE	High
0x2 0000	FLC2_MMR_ACCESS_ERROR	High

Table 8-6. HSM Error Aggregator

Error Type	Sync Bridge (MPU, DW, DR) Error	RAM, Bridges - DW Error	RAM, FRI, Bridges - DR Error	SSU	Priority
0x01	Security violation	Reserved	Reserved	Security violation (HSM_SSU_MMR_ACCESS_ERROR)	High
0x10	Access timeout error	Reserved	Reserved	Reserved	High
0x20	Access ACK error	Access ACK error	Access ACK error	Reserved	High
0x40	Uncorrectable error	Uncorrectable error	Uncorrectable error	Reserved	High
0x80	Correctable error	Reserved	Correctable error	Reserved	Low
0x4000	Access type = read	Reserved	Reserved	Reserved	- (Info only)
0x8000	Access type = write	Reserved	Reserved	Reserved	- (Info only)

8.7 Error Sources Information

This section lists all the error sources for each of the aggregators.

Table 8-7. Sources to CPUx Program Read (PR) Error Aggregator

Sources to CPUx PR Error Aggregator	Comments
CPUx	From respective CPU
CPUx ROM Controller	From respective ROM controller
FRI	CPU1-FRI1, CPU3-FRI2
LDA SRAM (LDA0-7)	Applicable to CPU1 and CPU2 only
LPA SRAM (LPA0-1)	Applicable to CPU1 and CPU2 only
CPA SRAM (CPA0-1)	Applicable to CPU1 and CPU3 only
CDA SRAM (CDA0-11)	Applicable to CPU1 and CPU3 only

Table 8-8. Sources to CPUx Data Read on Port 1/Port 2 (DR1/DR2) Error Aggregator

Sources to CPUx DR1/DR2 Error Aggregator	Comments
CPUx	From respective CPU
CPUx ROM Controller	From respective ROM Controller
FRI	Applicable to CPU1 and CPU3 only
M0	Applicable to CPU1 and CPU2 only
LDA SRAM (LDA0-7)	Applicable to CPU1 and CPU2 only
LPA SRAM (LPA0-1)	Applicable to CPU1 and CPU2 only
CPA SRAM (CPA0-1)	Applicable to CPU1 and CPU3 only
CDA SRAM (CDA0-11)	Applicable to CPU1 and CPU3 only
GAB1	Applicable to CPU3 only
GAB2	Applicable to CPU2 only
VBUSP_CPUx	From respective VBUSP_CPUx bridge
VBUSP_SSU	Applicable to CPU1-3
VBUSP_PROG	Applicable to CPU1 and CPU3 only
VBUS32_CONFIG	Applicable to CPU1-3
VBUSP_CONFIG	Applicable to CPU1-3
VBUS32_AP_CPUx	From respective VBUS32_AP_CPUx bridge
ADCWRAPPER	Applicable to CPU1-3
SOC_TO_HSM_Bridge	Applicable to CPU1-3
VBUSP_ETHERCAT	Applicable to CPU1-3

Table 8-8. Sources to CPUx Data Read on Port 1/Port 2 (DR1/DR2) Error Aggregator (continued)

Sources to CPUx DR1/DR2 Error Aggregator	Comments
VBUS32_ETHERCAT	Applicable to CPU1-3
PDI_ETHERCAT	Applicable to CPU1-3
VBUSP_frame0	Applicable to CPU1-3
VBUSP_frame1	Applicable to CPU1-3
VBUSP_frame2	Applicable to CPU1-3
VBUSP_frame3	Applicable to CPU1-3
VBUS32_frame0	Applicable to CPU1-3
VBUS32_frame1	Applicable to CPU1-3
VBUS32_frame2	Applicable to CPU1-3
VBUS32_frame3	Applicable to CPU1-3
VBUSP_EMIF	Applicable to CPU1-3
Data line buffer (DLB)	From respective CPU DLB

Table 8-9. Sources to CPUx DW (Data Write) Error Aggregator

Sources to CPUx DW Error Aggregator	Comments
CPUx	From respective CPU
M0	Applicable to CPU1 only
LDA SRAM (LDA0-7)	Applicable to CPU1 and CPU2 only
LPA SRAM (LPA0-1)	Applicable to CPU1 and CPU2 only
CPA SRAM (CPA0-1)	Applicable to CPU1 and CPU3 only
CDA SRAM (CDA0-11)	Applicable to CPU1 and CPU3 only
GAB1	Applicable to CPU3 only
GAB2	Applicable to CPU2 only
VBUSP_CPUx	From respective VBUSP_CPUx bridge
VBUSP_SSU	Applicable to CPU1-3
VBUSP_PROG	Applicable to CPU1 and CPU3 only
VBUS32_CONFIG	Applicable to CPU1-3
VBUSP_CONFIG	Applicable to CPU1-3
VBUS32_AP_CPUx	From respective VBUS32_AP_CPUx bridge
SOC_TO_HSM_Bridge	Applicable to CPU1-3
VBUSP_ETHERCAT	Applicable to CPU1-3
VBUS32_ETHERCAT	Applicable to CPU1-3
PDI_ETHERCAT	Applicable to CPU1-3
VBUSP_frame0	Applicable to CPU1-3
VBUSP_frame1	Applicable to CPU1-3
VBUSP_frame2	Applicable to CPU1-3
VBUSP_frame3	Applicable to CPU1-3
VBUS32_frame0	Applicable to CPU1-3
VBUS32_frame1	Applicable to CPU1-3
VBUS32_frame2	Applicable to CPU1-3
VBUS32_frame3	Applicable to CPU1-3
VBUSP_EMIF	Applicable to CPU1-3

Table 8-10. Sources to RTDMAx Data Read (DR) Error Aggregator

Sources to RTDMAx DR Error Aggregator	Comments
RTDMAx	From respective RTDMA
FRI	Applicable to RTDMA1 only
LDA SRAM (LDA0-7)	Applicable to RTDMA1 and RTDMA2
LPA SRAM (LPA0-1)	Applicable to RTDMA1 and RTDMA2
CPA SRAM (CPA0-1)	Applicable to RTDMA1 and RTDMA2
CDA SRAM (CDA0-11)	Applicable to RTDMA1 and RTDMA2
ADCWRAPPER	Applicable to RTDMA1 and RTDMA2
SOC_TO_HSM_Bridge	Applicable to RTDMA1 and RTDMA2
VBUSP_ETHERCAT	Applicable to RTDMA1 and RTDMA2
VBUS32_ETHERCAT	Applicable to RTDMA1 and RTDMA2
PDI_ETHERCAT	Applicable to RTDMA1 and RTDMA2
VBUSP_frame0	Applicable to RTDMA1 and RTDMA2
VBUSP_frame1	Applicable to RTDMA1 and RTDMA2
VBUSP_frame2	Applicable to RTDMA1 and RTDMA2
VBUSP_frame3	Applicable to RTDMA1 and RTDMA2
VBUS32_frame0	Applicable to RTDMA1 and RTDMA2
VBUS32_frame1	Applicable to RTDMA1 and RTDMA2
VBUS32_frame2	Applicable to RTDMA1 and RTDMA2
VBUS32_frame3	Applicable to RTDMA1 and RTDMA2
VBUSP_EMIF	Applicable to RTDMA1 and RTDMA2

Table 8-11. Sources to RTDMAx Data Write (DW) Error Aggregator

Sources to RTDMAx DW Error Aggregator	Comments
RTDMAx	From respective RTDMA
LDA SRAM (LDA0-7)	Applicable to RTDMA1 and RTDMA2
LPA SRAM (LPA0-1)	Applicable to RTDMA1 and RTDMA2
CPA SRAM (CPA0-1)	Applicable to RTDMA1 and RTDMA2
CDA SRAM (CDA0-11)	Applicable to RTDMA1 and RTDMA2
SOC_TO_HSM_Bridge	Applicable to RTDMA1 and RTDMA2
VBUSP_ETHERCAT	Applicable to RTDMA1 and RTDMA2
VBUS32_ETHERCAT	Applicable to RTDMA1 and RTDMA2
PDI_ETHERCAT	Applicable to RTDMA1 and RTDMA2
VBUSP_frame0	Applicable to RTDMA1 and RTDMA2
VBUSP_frame1	Applicable to RTDMA1 and RTDMA2
VBUSP_frame2	Applicable to RTDMA1 and RTDMA2
VBUSP_frame3	Applicable to RTDMA1 and RTDMA2
VBUS32_frame0	Applicable to RTDMA1 and RTDMA2
VBUS32_frame1	Applicable to RTDMA1 and RTDMA2
VBUS32_frame2	Applicable to RTDMA1 and RTDMA2
VBUS32_frame3	Applicable to RTDMA1 and RTDMA2
VBUSP_EMIF	Applicable to RTDMA1 and RTDMA2

Table 8-12. Sources to SSU Error Aggregator

Sources to SSU Error Aggregator	Comments
SSU	Errors from SSU

Table 8-13. Sources to CPUx Interrupt (INT) Error Aggregator

Sources to CPUx INT Error Aggregator	Comments
CPUx	From respective CPU
CPUx PIPE	From respective CPUx PIPE

Table 8-14. Sources to EtherCAT Error Aggregator

Sources to EtherCAT Error Aggregator	Comments
ETHERCAT Subsystem	From EtherCAT Subsystem Error_type fixed to 0x40

8.8 Software

8.8.1 ERROR_AGGREGATOR Registers to Driverlib Functions

Table 8-15. ERROR_AGGREGATOR Registers to Driverlib Functions

File	Driverlib Function
HSM_HIGHPRIO_ERROR_ADDRESS	
-	
HSM_LOWPRIO_ERROR_ADDRESS	
-	
HSM_ERROR_TYPE	
-	
HSM_ERROR_TYPE_FRC	
-	
HSM_ERROR_TYPE_CLR	
-	
CPU1_PR_HIGHPRIO_ERROR_ADDRESS	
-	
CPU1_PR_LOWPRIO_ERROR_ADDRESS	
-	
CPU1_PR_ERROR_TYPE	
-	
CPU1_PR_ERROR_TYPE_FRC	
-	
CPU1_PR_ERROR_TYPE_CLR	
-	
CPU1_PR_PC	
-	
CPU1_DR1_HIGHPRIO_ERROR_ADDRESS	
-	
CPU1_DR1_LOWPRIO_ERROR_ADDRESS	
-	
CPU1_DR1_ERROR_TYPE	

Table 8-15. ERROR_AGGREGATOR Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
CPU1_DR1_ERROR_TYPE_FRC	
-	
CPU1_DR1_ERROR_TYPE_CLR	
-	
CPU1_DR1_PC	
-	
CPU1_DR2_HIGHPRIO_ERROR_ADDRESS	
-	
CPU1_DR2_LOWPRIO_ERROR_ADDRESS	
-	
CPU1_DR2_ERROR_TYPE	
-	
CPU1_DR2_ERROR_TYPE_FRC	
-	
CPU1_DR2_ERROR_TYPE_CLR	
-	
CPU1_DR2_PC	
-	
CPU1_DW_HIGHPRIO_ERROR_ADDRESS	
-	
CPU1_DW_LOWPRIO_ERROR_ADDRESS	
-	
CPU1_DW_ERROR_TYPE	
-	
CPU1_DW_ERROR_TYPE_FRC	
-	
CPU1_DW_ERROR_TYPE_CLR	
-	
CPU1_DW_PC	
-	
CPU1_INT_HIGHPRIO_ERROR_ADDRESS	
-	
CPU1_INT_LOWPRIO_ERROR_ADDRESS	
-	
CPU1_INT_ERROR_TYPE	
-	
CPU1_INT_ERROR_TYPE_FRC	
-	
CPU1_INT_ERROR_TYPE_CLR	
-	
CPU1_INT_PC	
-	
CPU2_PR_HIGHPRIO_ERROR_ADDRESS	
-	

Table 8-15. ERROR_AGGREGATOR Registers to Driverlib Functions (continued)

File	Driverlib Function
CPU2_PR_LOWPRIO_ERROR_ADDRESS	
-	
CPU2_PR_ERROR_TYPE	
-	
CPU2_PR_ERROR_TYPE_FRC	
-	
CPU2_PR_ERROR_TYPE_CLR	
-	
CPU2_PR_PC	
-	
CPU2_DR1_HIGHPRIO_ERROR_ADDRESS	
-	
CPU2_DR1_LOWPRIO_ERROR_ADDRESS	
-	
CPU2_DR1_ERROR_TYPE	
-	
CPU2_DR1_ERROR_TYPE_FRC	
-	
CPU2_DR1_ERROR_TYPE_CLR	
-	
CPU2_DR1_PC	
-	
CPU2_DR2_HIGHPRIO_ERROR_ADDRESS	
-	
CPU2_DR2_LOWPRIO_ERROR_ADDRESS	
-	
CPU2_DR2_ERROR_TYPE	
-	
CPU2_DR2_ERROR_TYPE_FRC	
-	
CPU2_DR2_ERROR_TYPE_CLR	
-	
CPU2_DR2_PC	
-	
CPU2_DW_HIGHPRIO_ERROR_ADDRESS	
-	
CPU2_DW_LOWPRIO_ERROR_ADDRESS	
-	
CPU2_DW_ERROR_TYPE	
-	
CPU2_DW_ERROR_TYPE_FRC	
-	
CPU2_DW_ERROR_TYPE_CLR	
-	
CPU2_DW_PC	

Table 8-15. ERROR_AGGREGATOR Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
CPU2_INT_HIGHPRIO_ERROR_ADDRESS	
-	
CPU2_INT_LOWPRIO_ERROR_ADDRESS	
-	
CPU2_INT_ERROR_TYPE	
-	
CPU2_INT_ERROR_TYPE_FRC	
-	
CPU2_INT_ERROR_TYPE_CLR	
-	
CPU2_INT_PC	
-	
CPU3_PR_HIGHPRIO_ERROR_ADDRESS	
-	
CPU3_PR_LOWPRIO_ERROR_ADDRESS	
-	
CPU3_PR_ERROR_TYPE	
-	
CPU3_PR_ERROR_TYPE_FRC	
-	
CPU3_PR_ERROR_TYPE_CLR	
-	
CPU3_PR_PC	
-	
CPU3_DR1_HIGHPRIO_ERROR_ADDRESS	
-	
CPU3_DR1_LOWPRIO_ERROR_ADDRESS	
-	
CPU3_DR1_ERROR_TYPE	
-	
CPU3_DR1_ERROR_TYPE_FRC	
-	
CPU3_DR1_ERROR_TYPE_CLR	
-	
CPU3_DR1_PC	
-	
CPU3_DR2_HIGHPRIO_ERROR_ADDRESS	
-	
CPU3_DR2_LOWPRIO_ERROR_ADDRESS	
-	
CPU3_DR2_ERROR_TYPE	
-	
CPU3_DR2_ERROR_TYPE_FRC	
-	

Table 8-15. ERROR_AGGREGATOR Registers to Driverlib Functions (continued)

File	Driverlib Function
CPU3_DR2_ERROR_TYPE_CLR	
-	
CPU3_DR2_PC	
-	
CPU3_DW_HIGHPRIO_ERROR_ADDRESS	
-	
CPU3_DW_LOWPRIO_ERROR_ADDRESS	
-	
CPU3_DW_ERROR_TYPE	
-	
CPU3_DW_ERROR_TYPE_FRC	
-	
CPU3_DW_ERROR_TYPE_CLR	
-	
CPU3_DW_PC	
-	
CPU3_INT_HIGHPRIO_ERROR_ADDRESS	
-	
CPU3_INT_LOWPRIO_ERROR_ADDRESS	
-	
CPU3_INT_ERROR_TYPE	
-	
CPU3_INT_ERROR_TYPE_FRC	
-	
CPU3_INT_ERROR_TYPE_CLR	
-	
CPU3_INT_PC	
-	
RTDMA1_DR_HIGHPRIO_ERROR_ADDRESS	
-	
RTDMA1_DR_LOWPRIO_ERROR_ADDRESS	
-	
RTDMA1_DR_ERROR_TYPE	
-	
RTDMA1_DR_ERROR_TYPE_FRC	
-	
RTDMA1_DR_ERROR_TYPE_CLR	
-	
RTDMA1_DW_HIGHPRIO_ERROR_ADDRESS	
-	
RTDMA1_DW_LOWPRIO_ERROR_ADDRESS	
-	
RTDMA1_DW_ERROR_TYPE	
-	
RTDMA1_DW_ERROR_TYPE_FRC	

Table 8-15. ERROR_AGGREGATOR Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
RTDMA1_DW_ERROR_TYPE_CLR	
-	
RTDMA2_DR_HIGHPRIO_ERROR_ADDRESS	
-	
RTDMA2_DR_LOWPRIO_ERROR_ADDRESS	
-	
RTDMA2_DR_ERROR_TYPE	
-	
RTDMA2_DR_ERROR_TYPE_FRC	
-	
RTDMA2_DR_ERROR_TYPE_CLR	
-	
RTDMA2_DW_HIGHPRIO_ERROR_ADDRESS	
-	
RTDMA2_DW_LOWPRIO_ERROR_ADDRESS	
-	
RTDMA2_DW_ERROR_TYPE	
-	
RTDMA2_DW_ERROR_TYPE_FRC	
-	
RTDMA2_DW_ERROR_TYPE_CLR	
-	
SSU_HIGHPRIO_ERROR_ADDRESS	
-	
SSU_ERROR_TYPE	
-	
SSU_ERROR_TYPE_FRC	
-	
SSU_ERROR_TYPE_CLR	
-	
ETHERCAT_HIGHPRIO_ERROR_ADDRESS	
-	
ETHERCAT_ERROR_TYPE	
-	
ETHERCAT_ERROR_TYPE_FRC	
-	
ETHERCAT_ERROR_TYPE_CLR	
-	

8.9 ERRORAGGREGATOR Registers

This Section describes the ERRORAGGREGATOR Registers.

8.9.1 ERRORAGGREGATOR Base Address Table

Table 8-16. ERRORAGGREGATOR Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
HSM_ERROR_AGGREGATOR_CONFIG_REGS	HSMERRORAGGREGATOR_BASE	0x3012_0000	-	-	-	-	-	-	YES	YES
ERROR_AGGREGATOR_CONFIG_REGS	ERRORAGGREGATOR_BASE	0x6008_C000	YES	YES	YES	YES	YES	YES	-	YES

8.9.2 HSM_ERROR_AGGREGATOR_CONFIG_REGS Registers

Table 8-17 lists the memory-mapped registers for the HSM_ERROR_AGGREGATOR_CONFIG_REGS registers. All register offset addresses not listed in Table 8-17 should be considered as reserved locations and the register contents should not be modified.

Table 8-17. HSM_ERROR_AGGREGATOR_CONFIG_REGS Registers

Offset	Acronym	Register Name	Protection
0h	HSM_HIGHPRIO_ERROR_ADDRESS	High Priority Error address register	
4h	HSM_LOWPRIO_ERROR_ADDRESS	Low Priority Error address register	
8h	HSM_ERROR_TYPE	Error Type Register	
Ch	HSM_ERROR_TYPE_FRC	Error Type Force Register	
10h	HSM_ERROR_TYPE_CLR	Error Type Clear Register	

Complex bit access types are encoded to fit into small table cells. Table 8-18 shows the codes that are used for access types in this section.

Table 8-18. HSM_ERROR_AGGREGATOR_CONFIG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

8.9.2.1 HSM_HIGHPRIO_ERROR_ADDRESS Register (Offset = 0h) [Reset = 00000000h]

HSM_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-4 and described in Table 8-19.

Return to the Summary Table.

High Priority Error address register

Figure 8-4. HSM_HIGHPRIO_ERROR_ADDRESS Register

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

Table 8-19. HSM_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HSM_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on HSM access Reset type: PORESETn

8.9.2.2 HSM_LOWPRIO_ERROR_ADDRESS Register (Offset = 4h) [Reset = 00000000h]

HSM_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-5 and described in Table 8-20.

Return to the Summary Table.

Low Priority Error address register

Figure 8-5. HSM_LOWPRIO_ERROR_ADDRESS Register

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

Table 8-20. HSM_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HSM_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on HSM access Reset type: PORESETn

8.9.2.3 HSM_ERROR_TYPE Register (Offset = 8h) [Reset = 00000000h]

HSM_ERROR_TYPE is shown in Figure 8-6 and described in Table 8-21.

Return to the Summary Table.

Error Type Register

Figure 8-6. HSM_ERROR_TYPE Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0h	R-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
ACC_TYPE_WRITE	ACC_TYPE_READ	RESERVED					
R-0h	R-0h	R-0h					
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0h	R-0h	R-0h	R-0h	R-0h			R-0h

Table 8-21. HSM_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	High priority high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority high priority error set by software Reset type: PORESETn
29-16	RESERVED	R	0h	Reserved
15	ACC_TYPE_WRITE	R	0h	Access type write Reset type: PORESETn
14	ACC_TYPE_READ	R	0h	Access type Read Reset type: PORESETn
13-8	RESERVED	R	0h	Reserved
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R	0h	Security violation Reset type: PORESETn

8.9.2.4 HSM_ERROR_TYPE_FRC Register (Offset = Ch) [Reset = 00000000h]

HSM_ERROR_TYPE_FRC is shown in Figure 8-7 and described in Table 8-22.

Return to the Summary Table.

Error Type Force Register

Figure 8-7. HSM_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
ACC_TYPE_WRITE	ACC_TYPE_READ	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h			R-0/W1S-0h

Table 8-22. HSM_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	Force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-16	RESERVED	R	0h	Reserved
15	ACC_TYPE_WRITE	R-0/W1S	0h	Access type write flag force register 0 - No action 1 - force Access type write flag Reset type: PORESETn
14	ACC_TYPE_READ	R-0/W1S	0h	Access type read flag force register 0 - No action 1 - force Access type read flag Reset type: PORESETn
13-8	RESERVED	R	0h	Reserved
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn

Table 8-22. HSM_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1S	0h	Security violation flag force register 0 - No action 1 - force Security violation flag Reset type: PORESETn

8.9.2.5 HSM_ERROR_TYPE_CLR Register (Offset = 10h) [Reset = 00000000h]

HSM_ERROR_TYPE_CLR is shown in Figure 8-8 and described in Table 8-23.

Return to the Summary Table.

Error Type Clear Register

Figure 8-8. HSM_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
ACC_TYPE_WRITE	ACC_TYPE_READ	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0h			R-0/W1C-0h

Table 8-23. HSM_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-16	RESERVED	R	0h	Reserved
15	ACC_TYPE_WRITE	R-0/W1C	0h	Access type write flag clear register 0 - No action 1 - Clear Access type write flag Reset type: PORESETn
14	ACC_TYPE_READ	R-0/W1C	0h	Access type read flag clear register 0 - No action 1 - Clear Access type read flag Reset type: PORESETn
13-8	RESERVED	R	0h	Reserved
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn

Table 8-23. HSM_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1C	0h	Security violation flag clear register 0 - No action 1 - Clear Security violation flag Reset type: PORESETn

8.9.3 ERROR_AGGREGATOR_CONFIG_REGS Registers

Table 8-24 lists the memory-mapped registers for the ERROR_AGGREGATOR_CONFIG_REGS registers. All register offset addresses not listed in Table 8-24 should be considered as reserved locations and the register contents should not be modified.

Table 8-24. ERROR_AGGREGATOR_CONFIG_REGS Registers

Offset	Acronym	Register Name	Protection
0h	CPU1_PR_HIGHPRIO_ERROR_ADDRESS	CPU1 PR Error aggregator High Priority Error address register	
4h	CPU1_PR_LOWPRIO_ERROR_ADDRESS	CPU1 PR Error aggregator Low Priority Error address register	
8h	CPU1_PR_ERROR_TYPE	CPU1 PR Error aggregator Error Type Register	
Ch	CPU1_PR_ERROR_TYPE_FRC	CPU1 PR Error aggregator Error Type Force Register	
10h	CPU1_PR_ERROR_TYPE_CLR	CPU1 PR Error aggregator Error Type Clear Register	
14h	CPU1_PR_PC	CPU1 PR Error aggregator Register to store PC value at the first High priority error event	
40h	CPU1_DR1_HIGHPRIO_ERROR_ADDRESS	CPU1 DR1 Error aggregator High Priority Error address register	
44h	CPU1_DR1_LOWPRIO_ERROR_ADDRESS	CPU1 DR1 Error aggregator Low Priority Error address register	
48h	CPU1_DR1_ERROR_TYPE	CPU1 DR1 Error aggregator Error Type Register	
4Ch	CPU1_DR1_ERROR_TYPE_FRC	CPU1 DR1 Error aggregator Error Type Force Register	
50h	CPU1_DR1_ERROR_TYPE_CLR	CPU1 DR1 Error aggregator Error Type Clear Register	
54h	CPU1_DR1_PC	CPU1 DR1 Error aggregator Register to store PC value at the first High priority error event	
80h	CPU1_DR2_HIGHPRIO_ERROR_ADDRESS	CPU1 DR2 Error aggregator High Priority Error address register	
84h	CPU1_DR2_LOWPRIO_ERROR_ADDRESS	CPU1 DR2 Error aggregator Low Priority Error address register	
88h	CPU1_DR2_ERROR_TYPE	CPU1 DR2 Error aggregator Error Type Register	
8Ch	CPU1_DR2_ERROR_TYPE_FRC	CPU1 DR2 Error aggregator Error Type Force Register	
90h	CPU1_DR2_ERROR_TYPE_CLR	CPU1 DR2 Error aggregator Error Type Clear Register	
94h	CPU1_DR2_PC	CPU1 DR2 Error aggregator Register to store PC value at the first High priority error event	
C0h	CPU1_DW_HIGHPRIO_ERROR_ADDRESS	CPU1 DW Error aggregator High Priority Error address register	
C4h	CPU1_DW_LOWPRIO_ERROR_ADDRESS	CPU1 DW Error aggregator Low Priority Error address register	
C8h	CPU1_DW_ERROR_TYPE	CPU1 DW Error aggregator Error Type Register	
CCh	CPU1_DW_ERROR_TYPE_FRC	CPU1 DW Error aggregator Error Type Force Register	
D0h	CPU1_DW_ERROR_TYPE_CLR	CPU1 DW Error aggregator Error Type Clear Register	
D4h	CPU1_DW_PC	CPU1 DW Error aggregator Register to store PC value at the first High priority error event	
100h	CPU1_INT_HIGHPRIO_ERROR_ADDRESS	CPU1 INT Error aggregator High Priority Error address register	
104h	CPU1_INT_LOWPRIO_ERROR_ADDRESS	CPU1 INT Error aggregator Low Priority Error address register	
108h	CPU1_INT_ERROR_TYPE	CPU1 INT Error aggregator Error Type Register	
10Ch	CPU1_INT_ERROR_TYPE_FRC	CPU1 INT Error aggregator Error Type Force Register	
110h	CPU1_INT_ERROR_TYPE_CLR	CPU1 INT Error aggregator Error Type Clear Register	
114h	CPU1_INT_PC	CPU1 INT Error aggregator Register to store PC value at the first High priority error event	

Table 8-24. ERROR_AGGREGATOR_CONFIG_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
140h	CPU2_PR_HIGHPRIO_ERROR_ADDRESS	CPU2 PR Error aggregator High Priority Error address register	
144h	CPU2_PR_LOWPRIO_ERROR_ADDRESS	CPU2 PR Error aggregator Low Priority Error address register	
148h	CPU2_PR_ERROR_TYPE	CPU2 PR Error aggregator Error Type Register	
14Ch	CPU2_PR_ERROR_TYPE_FRC	CPU2 PR Error aggregator Error Type Force Register	
150h	CPU2_PR_ERROR_TYPE_CLR	CPU2 PR Error aggregator Error Type Clear Register	
154h	CPU2_PR_PC	CPU2 PR Error aggregator Register to store PC value at the first High priority error event	
180h	CPU2_DR1_HIGHPRIO_ERROR_ADDRESS	CPU2 DR1 Error aggregator High Priority Error address register	
184h	CPU2_DR1_LOWPRIO_ERROR_ADDRESS	CPU2 DR1 Error aggregator Low Priority Error address register	
188h	CPU2_DR1_ERROR_TYPE	CPU2 DR1 Error aggregator Error Type Register	
18Ch	CPU2_DR1_ERROR_TYPE_FRC	CPU2 DR1 Error aggregator Error Type Force Register	
190h	CPU2_DR1_ERROR_TYPE_CLR	CPU2 DR1 Error aggregator Error Type Clear Register	
194h	CPU2_DR1_PC	CPU2 DR1 Error aggregator Register to store PC value at the first High priority error event	
1C0h	CPU2_DR2_HIGHPRIO_ERROR_ADDRESS	CPU2 DR2 Error aggregator High Priority Error address register	
1C4h	CPU2_DR2_LOWPRIO_ERROR_ADDRESS	CPU2 DR2 Error aggregator Low Priority Error address register	
1C8h	CPU2_DR2_ERROR_TYPE	CPU2 DR2 Error aggregator Error Type Register	
1CCh	CPU2_DR2_ERROR_TYPE_FRC	CPU2 DR2 Error aggregator Error Type Force Register	
1D0h	CPU2_DR2_ERROR_TYPE_CLR	CPU2 DR2 Error aggregator Error Type Clear Register	
1D4h	CPU2_DR2_PC	CPU2 DR2 Error aggregator Register to store PC value at the first High priority error event	
200h	CPU2_DW_HIGHPRIO_ERROR_ADDRESS	CPU2 DW Error aggregator High Priority Error address register	
204h	CPU2_DW_LOWPRIO_ERROR_ADDRESS	CPU2 DW Error aggregator Low Priority Error address register	
208h	CPU2_DW_ERROR_TYPE	CPU2 DW Error aggregator Error Type Register	
20Ch	CPU2_DW_ERROR_TYPE_FRC	CPU2 DW Error aggregator Error Type Force Register	
210h	CPU2_DW_ERROR_TYPE_CLR	CPU2 DW Error aggregator Error Type Clear Register	
214h	CPU2_DW_PC	CPU2 DW Error aggregator Register to store PC value at the first High priority error event	
240h	CPU2_INT_HIGHPRIO_ERROR_ADDRESS	CPU2 INT Error aggregator High Priority Error address register	
244h	CPU2_INT_LOWPRIO_ERROR_ADDRESS	CPU2 INT Error aggregator Low Priority Error address register	
248h	CPU2_INT_ERROR_TYPE	CPU2 INT Error aggregator Error Type Register	
24Ch	CPU2_INT_ERROR_TYPE_FRC	CPU2 INT Error aggregator Error Type Force Register	
250h	CPU2_INT_ERROR_TYPE_CLR	CPU2 INT Error aggregator Error Type Clear Register	
254h	CPU2_INT_PC	CPU2 INT Error aggregator Register to store PC value at the first High priority error event	
280h	CPU3_PR_HIGHPRIO_ERROR_ADDRESS	CPU3 PR Error aggregator High Priority Error address register	
284h	CPU3_PR_LOWPRIO_ERROR_ADDRESS	CPU3 PR Error aggregator Low Priority Error address register	
288h	CPU3_PR_ERROR_TYPE	CPU3 PR Error aggregator Error Type Register	

Table 8-24. ERROR_AGGREGATOR_CONFIG_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
28Ch	CPU3_PR_ERROR_TYPE_FRC	CPU3 PR Error aggregator Error Type Force Register	
290h	CPU3_PR_ERROR_TYPE_CLR	CPU3 PR Error aggregator Error Type Clear Register	
294h	CPU3_PR_PC	CPU3 PR Error aggregator Register to store PC value at the first High priority error event	
2C0h	CPU3_DR1_HIGHPRIO_ERROR_ADDRESS	CPU3 DR1 Error aggregator High Priority Error address register	
2C4h	CPU3_DR1_LOWPRIO_ERROR_ADDRESS	CPU3 DR1 Error aggregator Low Priority Error address register	
2C8h	CPU3_DR1_ERROR_TYPE	CPU3 DR1 Error aggregator Error Type Register	
2CCh	CPU3_DR1_ERROR_TYPE_FRC	CPU3 DR1 Error aggregator Error Type Force Register	
2D0h	CPU3_DR1_ERROR_TYPE_CLR	CPU3 DR1 Error aggregator Error Type Clear Register	
2D4h	CPU3_DR1_PC	CPU3 DR1 Error aggregator Register to store PC value at the first High priority error event	
300h	CPU3_DR2_HIGHPRIO_ERROR_ADDRESS	CPU3 DR2 Error aggregator High Priority Error address register	
304h	CPU3_DR2_LOWPRIO_ERROR_ADDRESS	CPU3 DR2 Error aggregator Low Priority Error address register	
308h	CPU3_DR2_ERROR_TYPE	CPU3 DR2 Error aggregator Error Type Register	
30Ch	CPU3_DR2_ERROR_TYPE_FRC	CPU3 DR2 Error aggregator Error Type Force Register	
310h	CPU3_DR2_ERROR_TYPE_CLR	CPU3 DR2 Error aggregator Error Type Clear Register	
314h	CPU3_DR2_PC	CPU3 DR2 Error aggregator Register to store PC value at the first High priority error event	
340h	CPU3_DW_HIGHPRIO_ERROR_ADDRESS	CPU3 DW Error aggregator High Priority Error address register	
344h	CPU3_DW_LOWPRIO_ERROR_ADDRESS	CPU3 DW Error aggregator Low Priority Error address register	
348h	CPU3_DW_ERROR_TYPE	CPU3 DW Error aggregator Error Type Register	
34Ch	CPU3_DW_ERROR_TYPE_FRC	CPU3 DW Error aggregator Error Type Force Register	
350h	CPU3_DW_ERROR_TYPE_CLR	CPU3 DW Error aggregator Error Type Clear Register	
354h	CPU3_DW_PC	CPU3 DW Error aggregator Register to store PC value at the first High priority error event	
380h	CPU3_INT_HIGHPRIO_ERROR_ADDRESS	CPU3 INT Error aggregator High Priority Error address register	
384h	CPU3_INT_LOWPRIO_ERROR_ADDRESS	CPU3 INT Error aggregator Low Priority Error address register	
388h	CPU3_INT_ERROR_TYPE	CPU3 INT Error aggregator Error Type Register	
38Ch	CPU3_INT_ERROR_TYPE_FRC	CPU3 INT Error aggregator Error Type Force Register	
390h	CPU3_INT_ERROR_TYPE_CLR	CPU3 INT Error aggregator Error Type Clear Register	
394h	CPU3_INT_PC	CPU3 INT Error aggregator Register to store PC value at the first High priority error event	
780h	RTDMA1_DR_HIGHPRIO_ERROR_ADDRESS	RTDMA1 DR Error aggregator High Priority Error address register	
784h	RTDMA1_DR_LOWPRIO_ERROR_ADDRESS	RTDMA1 DR Error aggregator Low Priority Error address register	
788h	RTDMA1_DR_ERROR_TYPE	RTDMA1 DR Error aggregator Error Type Register	
78Ch	RTDMA1_DR_ERROR_TYPE_FRC	RTDMA1 DR Error aggregator Error Type Force Register	
790h	RTDMA1_DR_ERROR_TYPE_CLR	RTDMA1 DR Error aggregator Error Type Clear Register	
7C0h	RTDMA1_DW_HIGHPRIO_ERROR_ADDRESS	RTDMA1 DW Error aggregator High Priority Error address register	

Table 8-24. ERROR_AGGREGATOR_CONFIG_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
7C4h	RTDMA1_DW_LOWPRIO_ERROR_ADDRESS	RTDMA1 DW Error aggregator Low Priority Error address register	
7C8h	RTDMA1_DW_ERROR_TYPE	RTDMA1 DW Error aggregator Error Type Register	
7CCh	RTDMA1_DW_ERROR_TYPE_FRC	RTDMA1 DW Error aggregator Error Type Force Register	
7D0h	RTDMA1_DW_ERROR_TYPE_CLR	RTDMA1 DW Error aggregator Error Type Clear Register	
800h	RTDMA2_DR_HIGHPRIO_ERROR_ADDRESS	RTDMA2 DR Error aggregator High Priority Error address register	
804h	RTDMA2_DR_LOWPRIO_ERROR_ADDRESS	RTDMA2 DR Error aggregator Low Priority Error address register	
808h	RTDMA2_DR_ERROR_TYPE	RTDMA2 DR Error aggregator Error Type Register	
80Ch	RTDMA2_DR_ERROR_TYPE_FRC	RTDMA2 DR Error aggregator Error Type Force Register	
810h	RTDMA2_DR_ERROR_TYPE_CLR	RTDMA2 DR Error aggregator Error Type Clear Register	
840h	RTDMA2_DW_HIGHPRIO_ERROR_ADDRESS	RTDMA2 DW Error aggregator High Priority Error address register	
844h	RTDMA2_DW_LOWPRIO_ERROR_ADDRESS	RTDMA2 DW Error aggregator Low Priority Error address register	
848h	RTDMA2_DW_ERROR_TYPE	RTDMA2 DW Error aggregator Error Type Register	
84Ch	RTDMA2_DW_ERROR_TYPE_FRC	RTDMA2 DW Error aggregator Error Type Force Register	
850h	RTDMA2_DW_ERROR_TYPE_CLR	RTDMA2 DW Error aggregator Error Type Clear Register	
880h	SSU_HIGHPRIO_ERROR_ADDRESS	SSU Error aggregator High Priority Error address register	
888h	SSU_ERROR_TYPE	SSU Error aggregator Error Type Register	
88Ch	SSU_ERROR_TYPE_FRC	SSU Error aggregator Error Type Force Register	
890h	SSU_ERROR_TYPE_CLR	SSU Error aggregator Error Type Clear Register	
8C0h	ETHERCAT_HIGHPRIO_ERROR_ADDRESS	ETHERCAT Error aggregator High Priority Error address register	
8C8h	ETHERCAT_ERROR_TYPE	ETHERCAT Error aggregator Error Type Register	
8CCh	ETHERCAT_ERROR_TYPE_FRC	ETHERCAT Error aggregator Error Type Force Register	
8D0h	ETHERCAT_ERROR_TYPE_CLR	ETHERCAT Error aggregator Error Type Clear Register	

Complex bit access types are encoded to fit into small table cells. Table 8-25 shows the codes that are used for access types in this section.

Table 8-25. ERROR_AGGREGATOR_CONFIG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		

Table 8-25. ERROR_AGGREGATOR_CONFIG_REGS Access Type Codes (continued)

Access Type	Code	Description
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

8.9.3.1 CPU1_PR_HIGHPRIO_ERROR_ADDRESS Register (Offset = 0h) [Reset = 00000000h]

CPU1_PR_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-9 and described in Table 8-26.

Return to the Summary Table.

CPU1 PR Error aggregator High Priority Error address register

Figure 8-9. CPU1_PR_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU1_PR_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-26. CPU1_PR_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU1_PR_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on CPU1_PR access Reset type: PORESETn

8.9.3.2 CPU1_PR_LOWPRIO_ERROR_ADDRESS Register (Offset = 4h) [Reset = 00000000h]

CPU1_PR_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-10 and described in Table 8-27.

Return to the Summary Table.

CPU1 PR Error aggregator Low Priority Error address register

Figure 8-10. CPU1_PR_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU1_PR_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-27. CPU1_PR_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU1_PR_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on CPU1_PR access Reset type: PORESETn

8.9.3.3 CPU1_PR_ERROR_TYPE Register (Offset = 8h) [Reset = 00000000h]

CPU1_PR_ERROR_TYPE is shown in Figure 8-11 and described in Table 8-28.

Return to the Summary Table.

CPU1 PR Error aggregator Error Type Register

Figure 8-11. CPU1_PR_ERROR_TYPE Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0h	R-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				INSTRUCTION_TIMEOUT	ILLEGAL_INSTRUCTION	SW_BREAKPOINT_ERR	WARN_PSP_ERR
R-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	MAX_PSP_ERR	SEC_EXIT_VIO	SEC_ENTRY_VIO	INSTR_SECURITY_VIO
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 8-28. CPU1_PR_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority set by software Reset type: PORESETn
29-12	RESERVED	R	0h	Reserved
11	INSTRUCTION_TIMEOUT	R	0h	Instruction timeout error Reset type: PORESETn
10	ILLEGAL_INSTRUCTION	R	0h	Illegal instruction error Reset type: PORESETn
9	SW_BREAKPOINT_ERR	R	0h	Software brakepoint error Reset type: PORESETn
8	WARN_PSP_ERR	R	0h	Warn PSP error Reset type: PORESETn
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3	MAX_PSP_ERR	R	0h	MAX PSP error Reset type: PORESETn
2	SEC_EXIT_VIO	R	0h	Secure exit error Reset type: PORESETn

Table 8-28. CPU1_PR_ERROR_TYPE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	SEC_ENTRY_VIO	R	0h	Secure entry and linear code crossing LINK, STACK, ZONE error. Reset type: PORESETn
0	INSTR_SECURITY_VIO	R	0h	Instruction packet security violation. Instruction packet crossed LINK, STACK, ZONE boundary. Reset type: PORESETn

8.9.3.4 CPU1_PR_ERROR_TYPE_FRC Register (Offset = Ch) [Reset = 00000000h]

CPU1_PR_ERROR_TYPE_FRC is shown in Figure 8-12 and described in Table 8-29.

Return to the Summary Table.

CPU1 PR Error aggregator Error Type Force Register

Figure 8-12. CPU1_PR_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				INSTRUCTION_TIMEOUT	ILLEGAL_INSTRUCTION	SW_BREAKPOINT_ERR	WARN_PSP_ERR
R-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	MAX_PSP_ERR	SEC_EXIT_VIO	SEC_ENTRY_VIO	INSTR_SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 8-29. CPU1_PR_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	Force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-12	RESERVED	R	0h	Reserved
11	INSTRUCTION_TIMEOUT	R-0/W1S	0h	Instruction timeout error flag force register 0 - No action 1 - force Instruction timeout flag Reset type: PORESETn
10	ILLEGAL_INSTRUCTION	R-0/W1S	0h	Illegal instruction error flag force register 0 - No action 1 - Illegal instruction error flag Reset type: PORESETn
9	SW_BREAKPOINT_ERR	R-0/W1S	0h	Software brakepoint error flag force register 0 - No action 1 - force Software brakepoint error flag Reset type: PORESETn
8	WARN_PSP_ERR	R-0/W1S	0h	WARN PSP violation flag force register 0 - No action 1 - force WARN PSP violation flag Reset type: PORESETn

Table 8-29. CPU1_PR_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3	MAX_PSP_ERR	R-0/W1S	0h	MAX PSP violation flag force register 0 - No action 1 - force MAX PSP violation flag Reset type: PORESETn
2	SEC_EXIT_VIO	R-0/W1S	0h	Secure exit violation flag force register 0 - No action 1 - force Secure exit violation flag Reset type: PORESETn
1	SEC_ENTRY_VIO	R-0/W1S	0h	Secure entry violation flag force register 0 - No action 1 - force Secure entry violation flag Reset type: PORESETn
0	INSTR_SECURITY_VIO	R-0/W1S	0h	Instruction packet security violation flag force register 0 - No action 1 - force Instruction packet security violation flag Reset type: PORESETn

8.9.3.5 CPU1_PR_ERROR_TYPE_CLR Register (Offset = 10h) [Reset = 00000000h]

CPU1_PR_ERROR_TYPE_CLR is shown in Figure 8-13 and described in Table 8-30.

Return to the Summary Table.

CPU1 PR Error aggregator Error Type Clear Register

Figure 8-13. CPU1_PR_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				INSTRUCTION_TIMEOUT	ILLEGAL_INSTRUCTION	SW_BREAKPOINT_ERR	WARN_PSP_ERR
R-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	MAX_PSP_ERR	SEC_EXIT_VIO	SEC_ENTRY_VIO	INSTR_SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 8-30. CPU1_PR_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-12	RESERVED	R	0h	Reserved
11	INSTRUCTION_TIMEOUT	R-0/W1C	0h	Instruction timeout error flag clear register 0 - No action 1 - clear Instruction timeout flag Reset type: PORESETn
10	ILLEGAL_INSTRUCTION	R-0/W1C	0h	Illegal instruction error flag clear register 0 - No action 1 - Illegal instruction error flag Reset type: PORESETn
9	SW_BREAKPOINT_ERR	R-0/W1C	0h	Software brakepoint error flag clear register 0 - No action 1 - clear Software brakepoint error flag Reset type: PORESETn
8	WARN_PSP_ERR	R-0/W1C	0h	WARN PSP violation flag clear register 0 - No action 1 - Clear WARN PSP violation flag Reset type: PORESETn

Table 8-30. CPU1_PR_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3	MAX_PSP_ERR	R-0/W1C	0h	MAX PSP violation flag clear register 0 - No action 1 - Clear MAX PSP violation flag Reset type: PORESETn
2	SEC_EXIT_VIO	R-0/W1C	0h	Secure exit violation flag clear register 0 - No action 1 - Clear Secure exit violation flag Reset type: PORESETn
1	SEC_ENTRY_VIO	R-0/W1C	0h	Secure entry violation flag clear register 0 - No action 1 - Clear Secure entry violation flag Reset type: PORESETn
0	INSTR_SECURITY_VIO	R-0/W1C	0h	Instruction packet security violation flag clear register 0 - No action 1 - Clear Instruction packet security violation flag Reset type: PORESETn

8.9.3.6 CPU1_PR_PC Register (Offset = 14h) [Reset = 00000000h]

CPU1_PR_PC is shown in Figure 8-14 and described in Table 8-31.

Return to the Summary Table.

CPU1 PR Error aggregator Register to store PC value at the first High priority error event

Figure 8-14. CPU1_PR_PC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU1_PR_PC																															
R-0h																															

Table 8-31. CPU1_PR_PC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU1_PR_PC	R	0h	CPPU1 PC at which first High priority error occurred on CPU1_PR access Reset type: PORESETn

8.9.3.7 CPU1_DR1_HIGHPRIO_ERROR_ADDRESS Register (Offset = 40h) [Reset = 00000000h]

CPU1_DR1_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-15 and described in Table 8-32.

Return to the Summary Table.

CPU1 DR1 Error aggregator High Priority Error address register

Figure 8-15. CPU1_DR1_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU1_DR1_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-32. CPU1_DR1_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU1_DR1_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on CPU1_DR1 access Reset type: PORESETn

8.9.3.8 CPU1_DR1_LOWPRIO_ERROR_ADDRESS Register (Offset = 44h) [Reset = 00000000h]

CPU1_DR1_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-16 and described in Table 8-33.

Return to the Summary Table.

CPU1 DR1 Error aggregator Low Priority Error address register

Figure 8-16. CPU1_DR1_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU1_DR1_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-33. CPU1_DR1_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU1_DR1_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on CPU1_DR1 access Reset type: PORESETn

8.9.3.9 CPU1_DR1_ERROR_TYPE Register (Offset = 48h) [Reset = 00000000h]

CPU1_DR1_ERROR_TYPE is shown in Figure 8-17 and described in Table 8-34.

Return to the Summary Table.

CPU1 DR1 Error aggregator Error Type Register

Figure 8-17. CPU1_DR1_ERROR_TYPE Register

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

Table 8-34. CPU1_DR1_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority high priority error set by software Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R	0h	Unaligned address error Reset type: PORESETn
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R	0h	Security violation Reset type: PORESETn

8.9.3.10 CPU1_DR1_ERROR_TYPE_FRC Register (Offset = 4Ch) [Reset = 00000000h]

CPU1_DR1_ERROR_TYPE_FRC is shown in Figure 8-18 and described in Table 8-35.

Return to the Summary Table.

CPU1 DR1 Error aggregator Error Type Force Register

Figure 8-18. CPU1_DR1_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1S-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h			R-0/W1S-0h

Table 8-35. CPU1_DR1_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1S	0h	Unaligned address error flag force register 0 - No action 1 - force Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn

Table 8-35. CPU1_DR1_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1S	0h	Security violation flag force register 0 - No action 1 - force Security violation flag Reset type: PORESETn

8.9.3.11 CPU1_DR1_ERROR_TYPE_CLR Register (Offset = 50h) [Reset = 00000000h]

CPU1_DR1_ERROR_TYPE_CLR is shown in Figure 8-19 and described in Table 8-36.

Return to the Summary Table.

CPU1 DR1 Error aggregator Error Type Clear Register

Figure 8-19. CPU1_DR1_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1C-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0h			R-0/W1C-0h

Table 8-36. CPU1_DR1_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1C	0h	Unaligned address error flag clear register 0 - No action 1 - Clear Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn

Table 8-36. CPU1_DR1_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1C	0h	Security violation flag clear register 0 - No action 1 - Clear Security violation flag Reset type: PORESETn

8.9.3.12 CPU1_DR1_PC Register (Offset = 54h) [Reset = 00000000h]

CPU1_DR1_PC is shown in Figure 8-20 and described in Table 8-37.

Return to the Summary Table.

CPU1 DR1 Error aggregator Register to store PC value at the first High priority error event

Figure 8-20. CPU1_DR1_PC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU1_DR1_PC																															
R-0h																															

Table 8-37. CPU1_DR1_PC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU1_DR1_PC	R	0h	CPU1 PC at which first High priority error occurred on CPU1_DR1 access Reset type: PORESETn

8.9.3.13 CPU1_DR2_HIGHPRIO_ERROR_ADDRESS Register (Offset = 80h) [Reset = 00000000h]

CPU1_DR2_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-21 and described in Table 8-38.

Return to the Summary Table.

CPU1 DR2 Error aggregator High Priority Error address register

Figure 8-21. CPU1_DR2_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU1_DR2_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-38. CPU1_DR2_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU1_DR2_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on CPU1_DR2 access Reset type: PORESETn

8.9.3.14 CPU1_DR2_LOWPRIO_ERROR_ADDRESS Register (Offset = 84h) [Reset = 00000000h]

CPU1_DR2_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-22 and described in Table 8-39.

Return to the Summary Table.

CPU1 DR2 Error aggregator Low Priority Error address register

Figure 8-22. CPU1_DR2_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU1_DR2_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-39. CPU1_DR2_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU1_DR2_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on CPU1_DR2 access Reset type: PORESETn

8.9.3.15 CPU1_DR2_ERROR_TYPE Register (Offset = 88h) [Reset = 00000000h]

CPU1_DR2_ERROR_TYPE is shown in Figure 8-23 and described in Table 8-40.

Return to the Summary Table.

CPU1 DR2 Error aggregator Error Type Register

Figure 8-23. CPU1_DR2_ERROR_TYPE Register

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

Table 8-40. CPU1_DR2_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority error set by software Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R	0h	Unaligned address error Reset type: PORESETn
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R	0h	Security violation Reset type: PORESETn

8.9.3.16 CPU1_DR2_ERROR_TYPE_FRC Register (Offset = 8Ch) [Reset = 00000000h]

CPU1_DR2_ERROR_TYPE_FRC is shown in Figure 8-24 and described in Table 8-41.

Return to the Summary Table.

CPU1 DR2 Error aggregator Error Type Force Register

Figure 8-24. CPU1_DR2_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1S-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h			R-0/W1S-0h

Table 8-41. CPU1_DR2_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1S	0h	Unaligned address error flag force register 0 - No action 1 - force Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn

Table 8-41. CPU1_DR2_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1S	0h	Security violation flag force register 0 - No action 1 - force Security violation flag Reset type: PORESETn

8.9.3.17 CPU1_DR2_ERROR_TYPE_CLR Register (Offset = 90h) [Reset = 00000000h]

CPU1_DR2_ERROR_TYPE_CLR is shown in Figure 8-25 and described in Table 8-42.

Return to the Summary Table.

CPU1 DR2 Error aggregator Error Type Clear Register

Figure 8-25. CPU1_DR2_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1C-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0h			R-0/W1C-0h

Table 8-42. CPU1_DR2_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1C	0h	Unaligned address error flag clear register 0 - No action 1 - Clear Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn

Table 8-42. CPU1_DR2_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1C	0h	Security violation flag clear register 0 - No action 1 - Clear Security violation flag Reset type: PORESETn

8.9.3.18 CPU1_DR2_PC Register (Offset = 94h) [Reset = 00000000h]

CPU1_DR2_PC is shown in Figure 8-26 and described in Table 8-43.

Return to the Summary Table.

CPU1 DR2 Error aggregator Register to store PC value at the first High priority error event

Figure 8-26. CPU1_DR2_PC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU1_DR2_PC																															
R-0h																															

Table 8-43. CPU1_DR2_PC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU1_DR2_PC	R	0h	CPU1 PC at which first High priority error occurred on CPU1_DR2 access Reset type: PORESETn

8.9.3.19 CPU1_DW_HIGHPRIO_ERROR_ADDRESS Register (Offset = C0h) [Reset = 00000000h]

CPU1_DW_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-27 and described in Table 8-44.

Return to the Summary Table.

CPU1 DW Error aggregator High Priority Error address register

Figure 8-27. CPU1_DW_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU1_DW_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-44. CPU1_DW_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU1_DW_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on CPU1_DW access Reset type: PORESETn

8.9.3.20 CPU1_DW_LOWPRIO_ERROR_ADDRESS Register (Offset = C4h) [Reset = 00000000h]

CPU1_DW_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-28 and described in Table 8-45.

Return to the Summary Table.

CPU1 DW Error aggregator Low Priority Error address register

Figure 8-28. CPU1_DW_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU1_DW_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-45. CPU1_DW_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU1_DW_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on CPU1_DW access Reset type: PORESETn

8.9.3.21 CPU1_DW_ERROR_TYPE Register (Offset = C8h) [Reset = 00000000h]

CPU1_DW_ERROR_TYPE is shown in Figure 8-29 and described in Table 8-46.

Return to the Summary Table.

CPU1 DW Error aggregator Error Type Register

Figure 8-29. CPU1_DW_ERROR_TYPE Register

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

Table 8-46. CPU1_DW_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority error set by software Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R	0h	Unaligned address error Reset type: PORESETn
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R	0h	Security violation Reset type: PORESETn

8.9.3.22 CPU1_DW_ERROR_TYPE_FRC Register (Offset = CCh) [Reset = 00000000h]

CPU1_DW_ERROR_TYPE_FRC is shown in Figure 8-30 and described in Table 8-47.

Return to the Summary Table.

CPU1 DW Error aggregator Error Type Force Register

Figure 8-30. CPU1_DW_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1S-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h			R-0/W1S-0h

Table 8-47. CPU1_DW_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1S	0h	Unaligned address error flag force register 0 - No action 1 - force Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn

Table 8-47. CPU1_DW_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1S	0h	Security violation flag force register 0 - No action 1 - force Security violation flag Reset type: PORESETn

8.9.3.23 CPU1_DW_ERROR_TYPE_CLR Register (Offset = D0h) [Reset = 00000000h]

CPU1_DW_ERROR_TYPE_CLR is shown in Figure 8-31 and described in Table 8-48.

Return to the Summary Table.

CPU1 DW Error aggregator Error Type Clear Register

Figure 8-31. CPU1_DW_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1C-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0h			R-0/W1C-0h

Table 8-48. CPU1_DW_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1C	0h	Unaligned address error flag clear register 0 - No action 1 - Clear Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn

Table 8-48. CPU1_DW_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1C	0h	Security violation flag clear register 0 - No action 1 - Clear Security violation flag Reset type: PORESETn

8.9.3.24 CPU1_DW_PC Register (Offset = D4h) [Reset = 00000000h]

CPU1_DW_PC is shown in Figure 8-32 and described in Table 8-49.

Return to the Summary Table.

CPU1 DW Error aggregator Register to store PC value at the first High priority error event

Figure 8-32. CPU1_DW_PC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU1_DW_PC																															
R-0h																															

Table 8-49. CPU1_DW_PC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU1_DW_PC	R	0h	CPU1 PC at which first High priority error occurred on CPU1_DW access Reset type: PORESETn

8.9.3.25 CPU1_INT_HIGHPRIO_ERROR_ADDRESS Register (Offset = 100h) [Reset = 00000000h]

CPU1_INT_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-33 and described in Table 8-50.

Return to the Summary Table.

CPU1 INT Error aggregator High Priority Error address register

Figure 8-33. CPU1_INT_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU1_INT_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-50. CPU1_INT_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU1_INT_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on CPU1_INT Reset type: PORESETn

8.9.3.26 CPU1_INT_LOWPRIO_ERROR_ADDRESS Register (Offset = 104h) [Reset = 00000000h]

CPU1_INT_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-34 and described in Table 8-51.

Return to the Summary Table.

CPU1 INT Error aggregator Low Priority Error address register

Figure 8-34. CPU1_INT_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU1_INT_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-51. CPU1_INT_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU1_INT_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on CPU1_INT Reset type: PORESETn

8.9.3.27 CPU1_INT_ERROR_TYPE Register (Offset = 108h) [Reset = 00000000h]

CPU1_INT_ERROR_TYPE is shown in Figure 8-35 and described in Table 8-52.

Return to the Summary Table.

CPU1 INT Error aggregator Error Type Register

Figure 8-35. CPU1_INT_ERROR_TYPE Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	PIPE_LOCK_KEY_ERR	REG_PARITY_DIAG_ERR	REG_PARITY_ERR	PIPE_SECURITY_VIO	MAXISP	WARNISP
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
VECT_UNCERR	VECT_CERR	NMI_CONTEXT_RESTORE_VECT_UNCERR	NMI_CONTEXT_RESTORE_VECT_CERR	NMI_INTERRUPT_RETURN_ERR	NMI_VECT_UNCERR	NMI_VECT_CERR	NMI_ISR_ENTRY_ERR
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
NMI_MAXISP_ERR	RESERVED	RTINT_CONTEXT_RESTORE_VECT_UNCERR	RTINT_CONTEXT_RESTORE_VECT_CERR	RTINT_INTERRUPT_RETURN_ERR	RTINT_VECT_UNCERR	RTINT_VECT_CERR	RTINT_ISR_ENTRY_ERR
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
INT_INTERRUPT_RETURN_ERR	INT_VECT_UNCERR	INT_VECT_CERR	INT_ISR_ENTRY_ERR	MAIN_INTERRUPT_RETURN_ERR	MAIN_VECT_UNCERR	MAIN_VECT_CERR	MAIN_ISR_ENTRY_ERR
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 8-52. CPU1_INT_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority error set by software Reset type: PORESETn
29	PIPE_LOCK_KEY_ERR	R	0h	PIPE Lock/key error Reset type: PORESETn
28	REG_PARITY_DIAG_ERR	R	0h	PIPE Register parity diag error Reset type: PORESETn
27	REG_PARITY_ERR	R	0h	PIPE Reg parity error Reset type: PORESETn
26	PIPE_SECURITY_VIO	R	0h	PIPE security violation Reset type: PORESETn
25	MAXISP	R	0h	maxisp Reset type: PORESETn
24	WARNISP	R	0h	warnisp Reset type: PORESETn
23	VECT_UNCERR	R	0h	Vector Uncorrectable error generated by PIPE on software read of vector register having uncorrectable error Reset type: PORESETn
22	VECT_CERR	R	0h	Vector Correctable error generated by PIPE on software read of vector register having correctable error Reset type: PORESETn

Table 8-52. CPU1_INT_ERROR_TYPE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	NMI_CONTEXT_RESTORE_VECT_UNCERR	R	0h	NMI Context restore Uncorrectable vector error Reset type: PORESETn
20	NMI_CONTEXT_RESTORE_VECT_CERR	R	0h	NMI Context restore Correctable vector error Reset type: PORESETn
19	NMI_INTERRUPT_RETURN_ERR	R	0h	NMI Interrupt return error Reset type: PORESETn
18	NMI_VECT_UNCERR	R	0h	NMI Uncorrectable vector error Reset type: PORESETn
17	NMI_VECT_CERR	R	0h	NMI Correctable vector error Reset type: PORESETn
16	NMI_ISR_ENTRY_ERR	R	0h	NMI ISR entry error Reset type: PORESETn
15	NMI_MAXISP_ERR	R	0h	NMI MAXISP error Reset type: PORESETn
14	RESERVED	R	0h	Reserved
13	RTINT_CONTEXT_RESTORE_VECT_UNCERR	R	0h	RTINT Context restore Uncorrectable vector error Reset type: PORESETn
12	RTINT_CONTEXT_RESTORE_VECT_CERR	R	0h	RTINT Context restore Correctable vector error Reset type: PORESETn
11	RTINT_INTERRUPT_RETURN_ERR	R	0h	RTINT Interrupt return error Reset type: PORESETn
10	RTINT_VECT_UNCERR	R	0h	RTINT Uncorrectable vector error Reset type: PORESETn
9	RTINT_VECT_CERR	R	0h	RTINT Correctable vector error Reset type: PORESETn
8	RTINT_ISR_ENTRY_ERR	R	0h	RTINT ISR entry error Reset type: PORESETn
7	INT_INTERRUPT_RETURN_ERR	R	0h	INT INTerrupt return error Reset type: PORESETn
6	INT_VECT_UNCERR	R	0h	INT Uncorrectable vector error Reset type: PORESETn
5	INT_VECT_CERR	R	0h	INT Correctable vector error Reset type: PORESETn
4	INT_ISR_ENTRY_ERR	R	0h	INT ISR entry error Reset type: PORESETn
3	MAIN_INTERRUPT_RETURN_ERR	R	0h	Main Interrupt return error Reset type: PORESETn
2	MAIN_VECT_UNCERR	R	0h	Main Uncorrectable vector error Reset type: PORESETn
1	MAIN_VECT_CERR	R	0h	Main Correctable vector error Reset type: PORESETn
0	MAIN_ISR_ENTRY_ERR	R	0h	Main ISR entry error Reset type: PORESETn

8.9.3.28 CPU1_INT_ERROR_TYPE_FRC Register (Offset = 10Ch) [Reset = 00000000h]

CPU1_INT_ERROR_TYPE_FRC is shown in Figure 8-36 and described in Table 8-53.

Return to the Summary Table.

CPU1 INT Error aggregator Error Type Force Register

Figure 8-36. CPU1_INT_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	PIPE_LOCK_KEY_ERR	REG_PARITY_DIAG_ERR	REG_PARITY_ERR	PIPE_SECURITY_VIO	MAXISP	WARNISP
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
VECT_UNCERR	VECT_CERR	NMI_CONTEXT_RESTORE_VECT_UNCERR	NMI_CONTEXT_RESTORE_VECT_CERR	NMI_INTERRUPT_RETURN_ERR	NMI_VECT_UNCERR	NMI_VECT_CERR	NMI_ISR_ENTRY_ERR
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
NMI_MAXISP_ERR	RESERVED	RTINT_CONTEXT_RESTORE_VECT_UNCERR	RTINT_CONTEXT_RESTORE_VECT_CERR	RTINT_INTERRUPT_RETURN_ERR	RTINT_VECT_UNCERR	RTINT_VECT_CERR	RTINT_ISR_ENTRY_ERR
R-0/W1S-0h	R-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
INT_INTERRUPT_RETURN_ERR	INT_VECT_UNCERR	INT_VECT_CERR	INT_ISR_ENTRY_ERR	MAIN_INTERRUPT_RETURN_ERR	MAIN_VECT_UNCERR	MAIN_VECT_CERR	MAIN_ISR_ENTRY_ERR
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 8-53. CPU1_INT_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29	PIPE_LOCK_KEY_ERR	R-0/W1S	0h	PIPE Lock key error force register 0 - No action 1 - force PIPE lock key error flag Reset type: PORESETn
28	REG_PARITY_DIAG_ERR	R-0/W1S	0h	PIPE Register parity diag error force register 0 - No action 1 - force MAX PSP violation flag Reset type: PORESETn
27	REG_PARITY_ERR	R-0/W1S	0h	PIPE Register parity error force register 0 - No action 1 - force PIPE Register parity errort error flag Reset type: PORESETn
26	PIPE_SECURITY_VIO	R-0/W1S	0h	PIPE security violation force register 0 - No action 1 - force PIPE security violation flag Reset type: PORESETn

Table 8-53. CPU1_INT_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	MAXISP	R-0/W1S	0h	Maxisp error flag force register 0 - No action 1 - force Maxisp error flag Reset type: PORESETn
24	WARNISP	R-0/W1S	0h	Warnisp error flag force register 0 - No action 1 - force Warnisp error flag Reset type: PORESETn
23	VECT_UNCERR	R-0/W1S	0h	Vector uncorrectable error force register 0 - No action 1 - force vector uncorrectable eror flag Reset type: PORESETn
22	VECT_CERR	R-0/W1S	0h	Vector correctable error force register 0 - No action 1 - force vector correctable error flag Reset type: PORESETn
21	NMI_CONTEXT_RESTORE_VECT_UNCERR	R-0/W1S	0h	NMI Context restore Uncorrectable vector error flag force register 0 - No action 1 - force NMI Context restore Uncorrectable vector error flag Reset type: PORESETn
20	NMI_CONTEXT_RESTORE_VECT_CERR	R-0/W1S	0h	NMI Context restore Correctable vector error flag force register 0 - No action 1 - force NMI Context restore Correctable vector error flag Reset type: PORESETn
19	NMI_INTERRUPT_RETURN_ERR	R-0/W1S	0h	NMI Interrupt return error flag force register 0 - No action 1 - force NMI Interrupt return error flag Reset type: PORESETn
18	NMI_VECT_UNCERR	R-0/W1S	0h	NMI uncorrectable vector error flag force register 0 - No action 1 - force NMI uncorrectable vector error flag Reset type: PORESETn
17	NMI_VECT_CERR	R-0/W1S	0h	NMI Correctable vector error flag force register 0 - No action 1 - force NMI Correctable vector error flag Reset type: PORESETn
16	NMI_ISR_ENTRY_ERR	R-0/W1S	0h	NMI ISR entry error flag force register 0 - No action 1 - NMI ISR entry error flag Reset type: PORESETn
15	NMI_MAXISP_ERR	R-0/W1S	0h	NMI MAXISP error flag force register 0 - No action 1 - NMI MAXISP error flag Reset type: PORESETn
14	RESERVED	R	0h	Reserved
13	RTINT_CONTEXT_RESTORE_VECT_UNCERR	R-0/W1S	0h	RTINT Context restore Uncorrectable vector error flag force register 0 - No action 1 - force RTINT Context restore Uncorrectable vector error flag Reset type: PORESETn
12	RTINT_CONTEXT_RESTORE_VECT_CERR	R-0/W1S	0h	RTINT Context restore Correctable vector error flag force register 0 - No action 1 - force RTINT Context restore Correctable vector error flag Reset type: PORESETn

Table 8-53. CPU1_INT_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	RTINT_INTERRUPT_RETURN_ERR	R-0/W1S	0h	RTINT Interrupt return error flag force register 0 - No action 1 - force RTINT Interrupt return error flag Reset type: PORESETn
10	RTINT_VECT_UNCERR	R-0/W1S	0h	RTINT uncorrectable vector error flag force register 0 - No action 1 - force RTINT uncorrectable vector error flag Reset type: PORESETn
9	RTINT_VECT_CERR	R-0/W1S	0h	RTINT Correctable vector error flag force register 0 - No action 1 - force RTINT Correctable vector error flag Reset type: PORESETn
8	RTINT_ISR_ENTRY_ERR	R-0/W1S	0h	RTINT ISR entry error flag force register 0 - No action 1 - RTINT ISR entry error flag Reset type: PORESETn
7	INT_INTERRUPT_RETURN_ERR	R-0/W1S	0h	INT Interrupt return error flag force register 0 - No action 1 - force INT Interrupt return error flag Reset type: PORESETn
6	INT_VECT_UNCERR	R-0/W1S	0h	INT uncorrectable vector error flag force register 0 - No action 1 - force INT uncorrectable vector error flag Reset type: PORESETn
5	INT_VECT_CERR	R-0/W1S	0h	INT Correctable vector error flag force register 0 - No action 1 - force INT Correctable vector error flag Reset type: PORESETn
4	INT_ISR_ENTRY_ERR	R-0/W1S	0h	INT ISR entry error flag force register 0 - No action 1 - INT ISR entry error flag Reset type: PORESETn
3	MAIN_INTERRUPT_RETURN_ERR	R-0/W1S	0h	Main Interrupt return error flag force register 0 - No action 1 - force Main Interrupt return error flag Reset type: PORESETn
2	MAIN_VECT_UNCERR	R-0/W1S	0h	Main uncorrectable vector error flag force register 0 - No action 1 - force Main uncorrectable vector error flag Reset type: PORESETn
1	MAIN_VECT_CERR	R-0/W1S	0h	Main Correctable vector error flag force register 0 - No action 1 - force Main Correctable vector error flag Reset type: PORESETn
0	MAIN_ISR_ENTRY_ERR	R-0/W1S	0h	Main ISR entry error flag force register 0 - No action 1 - Main ISR entry error flag Reset type: PORESETn

8.9.3.29 CPU1_INT_ERROR_TYPE_CLR Register (Offset = 110h) [Reset = 00000000h]

CPU1_INT_ERROR_TYPE_CLR is shown in Figure 8-37 and described in Table 8-54.

Return to the Summary Table.

CPU1 INT Error aggregator Error Type Clear Register

Figure 8-37. CPU1_INT_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	PIPE_LOCK_KEY_ERR	REG_PARITY_DIAG_ERR	REG_PARITY_ERR	PIPE_SECURITY_VIO	MAXISP	WARNISP
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
23	22	21	20	19	18	17	16
VECT_UNCERR	VECT_CERR	NMI_CONTEXT_RESTORE_VECT_UNCERR	NMI_CONTEXT_RESTORE_VECT_CERR	NMI_INTERRUPT_RETURN_ERR	NMI_VECT_UNCERR	NMI_VECT_CERR	NMI_ISR_ENTRY_ERR
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
15	14	13	12	11	10	9	8
NMI_MAXISP_ERR	RESERVED	RTINT_CONTEXT_RESTORE_VECT_UNCERR	RTINT_CONTEXT_RESTORE_VECT_CERR	RTINT_INTERRUPT_RETURN_ERR	RTINT_VECT_UNCERR	RTINT_VECT_CERR	RTINT_ISR_ENTRY_ERR
R-0/W1C-0h	R-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
INT_INTERRUPT_RETURN_ERR	INT_VECT_UNCERR	INT_VECT_CERR	INT_ISR_ENTRY_ERR	MAIN_INTERRUPT_RETURN_ERR	MAIN_VECT_UNCERR	MAIN_VECT_CERR	MAIN_ISR_ENTRY_ERR
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 8-54. CPU1_INT_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29	PIPE_LOCK_KEY_ERR	R-0/W1C	0h	PIPE Lock key error clear register 0 - No action 1 - clear PIPE lock key error flag Reset type: PORESETn
28	REG_PARITY_DIAG_ERR	R-0/W1C	0h	PIPE Register parity diag error clear register 0 - No action 1 - clear MAX PSP violation flag Reset type: PORESETn
27	REG_PARITY_ERR	R-0/W1C	0h	PIPE Register parity error clear register 0 - No action 1 - clear PIPE Register parity errort error flag Reset type: PORESETn
26	PIPE_SECURITY_VIO	R-0/W1C	0h	PIPE security violation clear register 0 - No action 1 - clear PIPE security violation flag Reset type: PORESETn

Table 8-54. CPU1_INT_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	MAXISP	R-0/W1C	0h	Maxisp error flag clear register 0 - No action 1 - Clear Maxisp error flag Reset type: PORESETn
24	WARNISP	R-0/W1C	0h	Warnisp error flag clear register 0 - No action 1 - Clear Warnisp error flag Reset type: PORESETn
23	VECT_UNCERR	R-0/W1C	0h	Vector uncorrectable error clear register 0 - No action 1 - Clear vector uncorrectable error flag Reset type: PORESETn
22	VECT_CERR	R-0/W1C	0h	Vector correctable error clear register 0 - No action 1 - Clear vector correctable error flag Reset type: PORESETn
21	NMI_CONTEXT_RESTORE_VECT_UNCERR	R-0/W1C	0h	NMI Context restore Uncorrectable vector error flag clear register 0 - No action 1 - clear NMI Context restore Uncorrectable vector error flag Reset type: PORESETn
20	NMI_CONTEXT_RESTORE_VECT_CERR	R-0/W1C	0h	NMI Context restore Correctable vector error flag clear register 0 - No action 1 - clear NMI Context restore Correctable vector error flag Reset type: PORESETn
19	NMI_INTERRUPT_RETURN_ERR	R-0/W1C	0h	NMI Interrupt return error flag clear register 0 - No action 1 - clear NMI Interrupt return error flag Reset type: PORESETn
18	NMI_VECT_UNCERR	R-0/W1C	0h	NMI uncorrectable vector error flag clear register 0 - No action 1 - clear NMI uncorrectable vector error flag Reset type: PORESETn
17	NMI_VECT_CERR	R-0/W1C	0h	NMI Correctable vector error flag clear register 0 - No action 1 - clear NMI Correctable vector error flag Reset type: PORESETn
16	NMI_ISR_ENTRY_ERR	R-0/W1C	0h	NMI ISR entry error flag clear register 0 - No action 1 - NMI ISR entry error flag Reset type: PORESETn
15	NMI_MAXISP_ERR	R-0/W1C	0h	NMI MAXISP error flag clear register 0 - No action 1 - NMI MAXISP error flag Reset type: PORESETn
14	RESERVED	R	0h	Reserved
13	RTINT_CONTEXT_RESTORE_VECT_UNCERR	R-0/W1C	0h	RTINT Context restore Uncorrectable vector error flag clear register 0 - No action 1 - clear RTINT Context restore Uncorrectable vector error flag Reset type: PORESETn
12	RTINT_CONTEXT_RESTORE_VECT_CERR	R-0/W1C	0h	RTINT Context restore Correctable vector error flag clear register 0 - No action 1 - clear RTINT Context restore Correctable vector error flag Reset type: PORESETn

Table 8-54. CPU1_INT_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	RTINT_INTERRUPT_RETURN_ERR	R-0/W1C	0h	RTINT Interrupt return error flag clear register 0 - No action 1 - clear RTINT Interrupt return error flag Reset type: PORESETn
10	RTINT_VECT_UNCERR	R-0/W1C	0h	RTINT uncorrectable vector error flag clear register 0 - No action 1 - clear RTINT uncorrectable vector error flag Reset type: PORESETn
9	RTINT_VECT_CERR	R-0/W1C	0h	RTINT Correctable vector error flag clear register 0 - No action 1 - clear RTINT Correctable vector error flag Reset type: PORESETn
8	RTINT_ISR_ENTRY_ERR	R-0/W1C	0h	RTINT ISR entry error flag clear register 0 - No action 1 - RTINT ISR entry error flag Reset type: PORESETn
7	INT_INTERRUPT_RETURN_ERR	R-0/W1C	0h	INT Interrupt return error flag clear register 0 - No action 1 - clear INT Interrupt return error flag Reset type: PORESETn
6	INT_VECT_UNCERR	R-0/W1C	0h	INT uncorrectable vector error flag clear register 0 - No action 1 - clear INT uncorrectable vector error flag Reset type: PORESETn
5	INT_VECT_CERR	R-0/W1C	0h	INT Correctable vector error flag clear register 0 - No action 1 - clear INT Correctable vector error flag Reset type: PORESETn
4	INT_ISR_ENTRY_ERR	R-0/W1C	0h	INT ISR entry error flag clear register 0 - No action 1 - INT ISR entry error flag Reset type: PORESETn
3	MAIN_INTERRUPT_RETURN_ERR	R-0/W1C	0h	Main Interrupt return error flag clear register 0 - No action 1 - clear Main Interrupt return error flag Reset type: PORESETn
2	MAIN_VECT_UNCERR	R-0/W1C	0h	Main uncorrectable vector error flag clear register 0 - No action 1 - clear Main uncorrectable vector error flag Reset type: PORESETn
1	MAIN_VECT_CERR	R-0/W1C	0h	Main Correctable vector error flag clear register 0 - No action 1 - clear Main Correctable vector error flag Reset type: PORESETn
0	MAIN_ISR_ENTRY_ERR	R-0/W1C	0h	Main ISR entry error flag clear register 0 - No action 1 - Main ISR entry error flag Reset type: PORESETn

8.9.3.30 CPU1_INT_PC Register (Offset = 114h) [Reset = 00000000h]

CPU1_INT_PC is shown in Figure 8-38 and described in Table 8-55.

Return to the Summary Table.

CPU1 INT Error aggregator Register to store PC value at the first High priority error event

Figure 8-38. CPU1_INT_PC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU1_INT_PC																															
R-0h																															

Table 8-55. CPU1_INT_PC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU1_INT_PC	R	0h	CPU1 PC at which first High priority error occurred on CPU1_INT Reset type: PORESETn

8.9.3.31 CPU2_PR_HIGHPRIO_ERROR_ADDRESS Register (Offset = 140h) [Reset = 00000000h]

CPU2_PR_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-39 and described in Table 8-56.

Return to the Summary Table.

CPU2 PR Error aggregator High Priority Error address register

Figure 8-39. CPU2_PR_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU2_PR_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-56. CPU2_PR_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU2_PR_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on CPU2_PR access Reset type: PORESETn

8.9.3.32 CPU2_PR_LOWPRIO_ERROR_ADDRESS Register (Offset = 144h) [Reset = 00000000h]

CPU2_PR_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-40 and described in Table 8-57.

Return to the Summary Table.

CPU2 PR Error aggregator Low Priority Error address register

Figure 8-40. CPU2_PR_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU2_PR_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-57. CPU2_PR_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU2_PR_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on CPU2_PR access Reset type: PORESETn

8.9.3.33 CPU2_PR_ERROR_TYPE Register (Offset = 148h) [Reset = 00000000h]

CPU2_PR_ERROR_TYPE is shown in Figure 8-41 and described in Table 8-58.

Return to the Summary Table.

CPU2 PR Error aggregator Error Type Register

Figure 8-41. CPU2_PR_ERROR_TYPE Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0h	R-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				INSTRUCTION_TIMEOUT	ILLEGAL_INSTRUCTION	SW_BREAKPOINT_ERR	WARN_PSP_ERR
R-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	MAX_PSP_ERR	SEC_EXIT_VIO	SEC_ENTRY_VIO	INSTR_SECURITY_VIO
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 8-58. CPU2_PR_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority set by software Reset type: PORESETn
29-12	RESERVED	R	0h	Reserved
11	INSTRUCTION_TIMEOUT	R	0h	Instruction timeout error Reset type: PORESETn
10	ILLEGAL_INSTRUCTION	R	0h	Illegal instruction error Reset type: PORESETn
9	SW_BREAKPOINT_ERR	R	0h	Software brakepoint error Reset type: PORESETn
8	WARN_PSP_ERR	R	0h	Warn PSP error Reset type: PORESETn
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3	MAX_PSP_ERR	R	0h	MAX PSP error Reset type: PORESETn
2	SEC_EXIT_VIO	R	0h	Secure exit error Reset type: PORESETn

Table 8-58. CPU2_PR_ERROR_TYPE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	SEC_ENTRY_VIO	R	0h	Secure entry and linear code crossing LINK, STACK, ZONE error. Reset type: PORESETn
0	INSTR_SECURITY_VIO	R	0h	Instruction packet security violation. Instruction packet crossed LINK, STACK, ZONE boundary. Reset type: PORESETn

8.9.3.34 CPU2_PR_ERROR_TYPE_FRC Register (Offset = 14Ch) [Reset = 00000000h]

CPU2_PR_ERROR_TYPE_FRC is shown in Figure 8-42 and described in Table 8-59.

Return to the Summary Table.

CPU2 PR Error aggregator Error Type Force Register

Figure 8-42. CPU2_PR_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				INSTRUCTION_TIMEOUT	ILLEGAL_INSTRUCTION	SW_BREAKPOINT_ERR	WARN_PSP_ERR
R-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	MAX_PSP_ERR	SEC_EXIT_VIO	SEC_ENTRY_VIO	INSTR_SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 8-59. CPU2_PR_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	Force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-12	RESERVED	R	0h	Reserved
11	INSTRUCTION_TIMEOUT	R-0/W1S	0h	Instruction timeout error flag force register 0 - No action 1 - force Instruction timeout flag Reset type: PORESETn
10	ILLEGAL_INSTRUCTION	R-0/W1S	0h	Illegal instruction error flag force register 0 - No action 1 - Illegal instruction error flag Reset type: PORESETn
9	SW_BREAKPOINT_ERR	R-0/W1S	0h	Software brakepoint error flag force register 0 - No action 1 - force Software brakepoint error flag Reset type: PORESETn
8	WARN_PSP_ERR	R-0/W1S	0h	WARN PSP violation flag force register 0 - No action 1 - force WARN PSP violation flag Reset type: PORESETn

Table 8-59. CPU2_PR_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3	MAX_PSP_ERR	R-0/W1S	0h	MAX PSP violation flag force register 0 - No action 1 - force MAX PSP violation flag Reset type: PORESETn
2	SEC_EXIT_VIO	R-0/W1S	0h	Secure exit violation flag force register 0 - No action 1 - force Secure exit violation flag Reset type: PORESETn
1	SEC_ENTRY_VIO	R-0/W1S	0h	Secure entry violation flag force register 0 - No action 1 - force Secure entry violation flag Reset type: PORESETn
0	INSTR_SECURITY_VIO	R-0/W1S	0h	Instruction packet security violation flag force register 0 - No action 1 - force Instruction packet security violation flag Reset type: PORESETn

8.9.3.35 CPU2_PR_ERROR_TYPE_CLR Register (Offset = 150h) [Reset = 00000000h]

CPU2_PR_ERROR_TYPE_CLR is shown in Figure 8-43 and described in Table 8-60.

Return to the Summary Table.

CPU2 PR Error aggregator Error Type Clear Register

Figure 8-43. CPU2_PR_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				INSTRUCTION_TIMEOUT	ILLEGAL_INSTRUCTION	SW_BREAKPOINT_ERR	WARN_PSP_ERR
R-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	MAX_PSP_ERR	SEC_EXIT_VIO	SEC_ENTRY_VIO	INSTR_SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 8-60. CPU2_PR_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-12	RESERVED	R	0h	Reserved
11	INSTRUCTION_TIMEOUT	R-0/W1C	0h	Instruction timeout error flag clear register 0 - No action 1 - clear Instruction timeout flag Reset type: PORESETn
10	ILLEGAL_INSTRUCTION	R-0/W1C	0h	Illegal instruction error flag clear register 0 - No action 1 - Illegal instruction error flag Reset type: PORESETn
9	SW_BREAKPOINT_ERR	R-0/W1C	0h	Software brakepoint error flag clear register 0 - No action 1 - clear Software brakepoint error flag Reset type: PORESETn
8	WARN_PSP_ERR	R-0/W1C	0h	WARN PSP violation flag clear register 0 - No action 1 - Clear WARN PSP violation flag Reset type: PORESETn

Table 8-60. CPU2_PR_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3	MAX_PSP_ERR	R-0/W1C	0h	MAX PSP violation flag clear register 0 - No action 1 - Clear MAX PSP violation flag Reset type: PORESETn
2	SEC_EXIT_VIO	R-0/W1C	0h	Secure exit violation flag clear register 0 - No action 1 - Clear Secure exit violation flag Reset type: PORESETn
1	SEC_ENTRY_VIO	R-0/W1C	0h	Secure entry violation flag clear register 0 - No action 1 - Clear Secure entry violation flag Reset type: PORESETn
0	INSTR_SECURITY_VIO	R-0/W1C	0h	Instruction packet security violation flag clear register 0 - No action 1 - Clear Instruction packet security violation flag Reset type: PORESETn

8.9.3.36 CPU2_PR_PC Register (Offset = 154h) [Reset = 00000000h]

CPU2_PR_PC is shown in Figure 8-44 and described in Table 8-61.

Return to the Summary Table.

CPU2 PR Error aggregator Register to store PC value at the first High priority error event

Figure 8-44. CPU2_PR_PC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU2_PR_PC																															
R-0h																															

Table 8-61. CPU2_PR_PC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU2_PR_PC	R	0h	CPPU2 PC at which first High priority error occurred on CPU2_PR access Reset type: PORESETn

8.9.3.37 CPU2_DR1_HIGHPRIO_ERROR_ADDRESS Register (Offset = 180h) [Reset = 00000000h]

CPU2_DR1_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-45 and described in Table 8-62.

Return to the Summary Table.

CPU2 DR1 Error aggregator High Priority Error address register

Figure 8-45. CPU2_DR1_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU2_DR1_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-62. CPU2_DR1_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU2_DR1_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on CPU2_DR1 access Reset type: PORESETn

8.9.3.38 CPU2_DR1_LOWPRIO_ERROR_ADDRESS Register (Offset = 184h) [Reset = 00000000h]

CPU2_DR1_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-46 and described in Table 8-63.

Return to the Summary Table.

CPU2 DR1 Error aggregator Low Priority Error address register

Figure 8-46. CPU2_DR1_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU2_DR1_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-63. CPU2_DR1_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU2_DR1_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on CPU2_DR1 access Reset type: PORESETn

8.9.3.39 CPU2_DR1_ERROR_TYPE Register (Offset = 188h) [Reset = 00000000h]

CPU2_DR1_ERROR_TYPE is shown in Figure 8-47 and described in Table 8-64.

Return to the Summary Table.

CPU2 DR1 Error aggregator Error Type Register

Figure 8-47. CPU2_DR1_ERROR_TYPE Register

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

Table 8-64. CPU2_DR1_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority high priority error set by software Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R	0h	Unaligned address error Reset type: PORESETn
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R	0h	Security violation Reset type: PORESETn

8.9.3.40 CPU2_DR1_ERROR_TYPE_FRC Register (Offset = 18Ch) [Reset = 00000000h]

CPU2_DR1_ERROR_TYPE_FRC is shown in Figure 8-48 and described in Table 8-65.

Return to the Summary Table.

CPU2 DR1 Error aggregator Error Type Force Register

Figure 8-48. CPU2_DR1_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1S-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h			R-0/W1S-0h

Table 8-65. CPU2_DR1_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1S	0h	Unaligned address error flag force register 0 - No action 1 - force Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn

Table 8-65. CPU2_DR1_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1S	0h	Security violation flag force register 0 - No action 1 - force Security violation flag Reset type: PORESETn

8.9.3.41 CPU2_DR1_ERROR_TYPE_CLR Register (Offset = 190h) [Reset = 00000000h]

CPU2_DR1_ERROR_TYPE_CLR is shown in Figure 8-49 and described in Table 8-66.

Return to the Summary Table.

CPU2 DR1 Error aggregator Error Type Clear Register

Figure 8-49. CPU2_DR1_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1C-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0h			R-0/W1C-0h

Table 8-66. CPU2_DR1_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1C	0h	Unaligned address error flag clear register 0 - No action 1 - Clear Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn

Table 8-66. CPU2_DR1_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1C	0h	Security violation flag clear register 0 - No action 1 - Clear Security violation flag Reset type: PORESETn

8.9.3.42 CPU2_DR1_PC Register (Offset = 194h) [Reset = 00000000h]

CPU2_DR1_PC is shown in Figure 8-50 and described in Table 8-67.

Return to the Summary Table.

CPU2 DR1 Error aggregator Register to store PC value at the first High priority error event

Figure 8-50. CPU2_DR1_PC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU2_DR1_PC																															
R-0h																															

Table 8-67. CPU2_DR1_PC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU2_DR1_PC	R	0h	CPU2 PC at which first High priority error occurred on CPU2_DR1 access Reset type: PORESETn

8.9.3.43 CPU2_DR2_HIGHPRIO_ERROR_ADDRESS Register (Offset = 1C0h) [Reset = 00000000h]

CPU2_DR2_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-51 and described in Table 8-68.

Return to the Summary Table.

CPU2 DR2 Error aggregator High Priority Error address register

Figure 8-51. CPU2_DR2_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU2_DR2_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-68. CPU2_DR2_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU2_DR2_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on CPU2_DR2 access Reset type: PORESETn

8.9.3.44 CPU2_DR2_LOWPRIO_ERROR_ADDRESS Register (Offset = 1C4h) [Reset = 00000000h]

CPU2_DR2_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-52 and described in Table 8-69.

Return to the Summary Table.

CPU2 DR2 Error aggregator Low Priority Error address register

Figure 8-52. CPU2_DR2_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU2_DR2_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-69. CPU2_DR2_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU2_DR2_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on CPU2_DR2 access Reset type: PORESETn

8.9.3.45 CPU2_DR2_ERROR_TYPE Register (Offset = 1C8h) [Reset = 00000000h]

CPU2_DR2_ERROR_TYPE is shown in Figure 8-53 and described in Table 8-70.

Return to the Summary Table.

CPU2 DR2 Error aggregator Error Type Register

Figure 8-53. CPU2_DR2_ERROR_TYPE Register

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

Table 8-70. CPU2_DR2_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority error set by software Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R	0h	Unaligned address error Reset type: PORESETn
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R	0h	Security violation Reset type: PORESETn

8.9.3.46 CPU2_DR2_ERROR_TYPE_FRC Register (Offset = 1CCh) [Reset = 00000000h]

CPU2_DR2_ERROR_TYPE_FRC is shown in Figure 8-54 and described in Table 8-71.

Return to the Summary Table.

CPU2 DR2 Error aggregator Error Type Force Register

Figure 8-54. CPU2_DR2_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1S-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h			R-0/W1S-0h

Table 8-71. CPU2_DR2_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1S	0h	Unaligned address error flag force register 0 - No action 1 - force Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn

Table 8-71. CPU2_DR2_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1S	0h	Security violation flag force register 0 - No action 1 - force Security violation flag Reset type: PORESETn

8.9.3.47 CPU2_DR2_ERROR_TYPE_CLR Register (Offset = 1D0h) [Reset = 00000000h]

CPU2_DR2_ERROR_TYPE_CLR is shown in Figure 8-55 and described in Table 8-72.

Return to the Summary Table.

CPU2 DR2 Error aggregator Error Type Clear Register

Figure 8-55. CPU2_DR2_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1C-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0h			R-0/W1C-0h

Table 8-72. CPU2_DR2_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1C	0h	Unaligned address error flag clear register 0 - No action 1 - Clear Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn

Table 8-72. CPU2_DR2_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1C	0h	Security violation flag clear register 0 - No action 1 - Clear Security violation flag Reset type: PORESETn

8.9.3.48 CPU2_DR2_PC Register (Offset = 1D4h) [Reset = 00000000h]

CPU2_DR2_PC is shown in Figure 8-56 and described in Table 8-73.

Return to the Summary Table.

CPU2 DR2 Error aggregator Register to store PC value at the first High priority error event

Figure 8-56. CPU2_DR2_PC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU2_DR2_PC																															
R-0h																															

Table 8-73. CPU2_DR2_PC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU2_DR2_PC	R	0h	CPU2 PC at which first High priority error occurred on CPU2_DR2 access Reset type: PORESETn

8.9.3.49 CPU2_DW_HIGHPRIO_ERROR_ADDRESS Register (Offset = 200h) [Reset = 00000000h]

CPU2_DW_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-57 and described in Table 8-74.

Return to the Summary Table.

CPU2 DW Error aggregator High Priority Error address register

Figure 8-57. CPU2_DW_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU2_DW_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-74. CPU2_DW_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU2_DW_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on CPU2_DW access Reset type: PORESETn

8.9.3.50 CPU2_DW_LOWPRIO_ERROR_ADDRESS Register (Offset = 204h) [Reset = 00000000h]

CPU2_DW_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-58 and described in Table 8-75.

Return to the Summary Table.

CPU2 DW Error aggregator Low Priority Error address register

Figure 8-58. CPU2_DW_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU2_DW_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-75. CPU2_DW_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU2_DW_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on CPU2_DW access Reset type: PORESETn

8.9.3.51 CPU2_DW_ERROR_TYPE Register (Offset = 208h) [Reset = 00000000h]

CPU2_DW_ERROR_TYPE is shown in Figure 8-59 and described in Table 8-76.

Return to the Summary Table.

CPU2 DW Error aggregator Error Type Register

Figure 8-59. CPU2_DW_ERROR_TYPE Register

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

Table 8-76. CPU2_DW_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority error set by software Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R	0h	Unaligned address error Reset type: PORESETn
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R	0h	Security violation Reset type: PORESETn

8.9.3.52 CPU2_DW_ERROR_TYPE_FRC Register (Offset = 20Ch) [Reset = 00000000h]

CPU2_DW_ERROR_TYPE_FRC is shown in Figure 8-60 and described in Table 8-77.

Return to the Summary Table.

CPU2 DW Error aggregator Error Type Force Register

Figure 8-60. CPU2_DW_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1S-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h			R-0/W1S-0h

Table 8-77. CPU2_DW_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1S	0h	Unaligned address error flag force register 0 - No action 1 - force Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn

Table 8-77. CPU2_DW_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1S	0h	Security violation flag force register 0 - No action 1 - force Security violation flag Reset type: PORESETn

8.9.3.53 CPU2_DW_ERROR_TYPE_CLR Register (Offset = 210h) [Reset = 00000000h]

CPU2_DW_ERROR_TYPE_CLR is shown in Figure 8-61 and described in Table 8-78.

Return to the Summary Table.

CPU2 DW Error aggregator Error Type Clear Register

Figure 8-61. CPU2_DW_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1C-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0h			R-0/W1C-0h

Table 8-78. CPU2_DW_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1C	0h	Unaligned address error flag clear register 0 - No action 1 - Clear Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn

Table 8-78. CPU2_DW_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1C	0h	Security violation flag clear register 0 - No action 1 - Clear Security violation flag Reset type: PORESETn

8.9.3.54 CPU2_DW_PC Register (Offset = 214h) [Reset = 00000000h]

CPU2_DW_PC is shown in Figure 8-62 and described in Table 8-79.

Return to the Summary Table.

CPU2 DW Error aggregator Register to store PC value at the first High priority error event

Figure 8-62. CPU2_DW_PC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU2_DW_PC																															
R-0h																															

Table 8-79. CPU2_DW_PC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU2_DW_PC	R	0h	CPU2 PC at which first High priority error occurred on CPU2_DW access Reset type: PORESETn

8.9.3.55 CPU2_INT_HIGHPRIO_ERROR_ADDRESS Register (Offset = 240h) [Reset = 00000000h]

CPU2_INT_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-63 and described in Table 8-80.

Return to the Summary Table.

CPU2 INT Error aggregator High Priority Error address register

Figure 8-63. CPU2_INT_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU2_INT_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-80. CPU2_INT_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU2_INT_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on CPU2_INT Reset type: PORESETn

8.9.3.56 CPU2_INT_LOWPRIO_ERROR_ADDRESS Register (Offset = 244h) [Reset = 00000000h]

CPU2_INT_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-64 and described in Table 8-81.

Return to the Summary Table.

CPU2 INT Error aggregator Low Priority Error address register

Figure 8-64. CPU2_INT_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU2_INT_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-81. CPU2_INT_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU2_INT_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on CPU2_INT Reset type: PORESETn

8.9.3.57 CPU2_INT_ERROR_TYPE Register (Offset = 248h) [Reset = 00000000h]

CPU2_INT_ERROR_TYPE is shown in Figure 8-65 and described in Table 8-82.

Return to the Summary Table.

CPU2 INT Error aggregator Error Type Register

Figure 8-65. CPU2_INT_ERROR_TYPE Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	PIPE_LOCK_KEY_ERR	REG_PARITY_DIAG_ERR	REG_PARITY_ERR	PIPE_SECURITY_VIO	MAXISP	WARNISP
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
VECT_UNCERR	VECT_CERR	NMI_CONTEXT_RESTORE_VECT_UNCERR	NMI_CONTEXT_RESTORE_VECT_CERR	NMI_INTERRUPT_RETURN_ERR	NMI_VECT_UNCERR	NMI_VECT_CERR	NMI_ISR_ENTRY_ERR
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
NMI_MAXISP_ERR	RESERVED	RTINT_CONTEXT_RESTORE_VECT_UNCERR	RTINT_CONTEXT_RESTORE_VECT_CERR	RTINT_INTERRUPT_RETURN_ERR	RTINT_VECT_UNCERR	RTINT_VECT_CERR	RTINT_ISR_ENTRY_ERR
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
INT_INTERRUPT_RETURN_ERR	INT_VECT_UNCERR	INT_VECT_CERR	INT_ISR_ENTRY_ERR	MAIN_INTERRUPT_RETURN_ERR	MAIN_VECT_UNCERR	MAIN_VECT_CERR	MAIN_ISR_ENTRY_ERR
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 8-82. CPU2_INT_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority error set by software Reset type: PORESETn
29	PIPE_LOCK_KEY_ERR	R	0h	PIPE Lock/key error Reset type: PORESETn
28	REG_PARITY_DIAG_ERR	R	0h	PIPE Register parity diag error Reset type: PORESETn
27	REG_PARITY_ERR	R	0h	PIPE Reg parity error Reset type: PORESETn
26	PIPE_SECURITY_VIO	R	0h	PIPE security violation Reset type: PORESETn
25	MAXISP	R	0h	maxisp Reset type: PORESETn
24	WARNISP	R	0h	warnisp Reset type: PORESETn
23	VECT_UNCERR	R	0h	Vector Uncorrectable error generated by PIPE on software read of vector register having uncorrectable error Reset type: PORESETn
22	VECT_CERR	R	0h	Vector Correctable error generated by PIPE on software read of vector register having correctable error Reset type: PORESETn

Table 8-82. CPU2_INT_ERROR_TYPE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	NMI_CONTEXT_RESTORE_VECT_UNCERR	R	0h	NMI Context restore Uncorrectable vector error Reset type: PORESETn
20	NMI_CONTEXT_RESTORE_VECT_CERR	R	0h	NMI Context restore Correctable vector error Reset type: PORESETn
19	NMI_INTERRUPT_RETURN_ERR	R	0h	NMI Interrupt return error Reset type: PORESETn
18	NMI_VECT_UNCERR	R	0h	NMI Uncorrectable vector error Reset type: PORESETn
17	NMI_VECT_CERR	R	0h	NMI Correctable vector error Reset type: PORESETn
16	NMI_ISR_ENTRY_ERR	R	0h	NMI ISR entry error Reset type: PORESETn
15	NMI_MAXISP_ERR	R	0h	NMI MAXISP error Reset type: PORESETn
14	RESERVED	R	0h	Reserved
13	RTINT_CONTEXT_RESTORE_VECT_UNCERR	R	0h	RTINT Context restore Uncorrectable vector error Reset type: PORESETn
12	RTINT_CONTEXT_RESTORE_VECT_CERR	R	0h	RTINT Context restore Correctable vector error Reset type: PORESETn
11	RTINT_INTERRUPT_RETURN_ERR	R	0h	RTINT Interrupt return error Reset type: PORESETn
10	RTINT_VECT_UNCERR	R	0h	RTINT Uncorrectable vector error Reset type: PORESETn
9	RTINT_VECT_CERR	R	0h	RTINT Correctable vector error Reset type: PORESETn
8	RTINT_ISR_ENTRY_ERR	R	0h	RTINT ISR entry error Reset type: PORESETn
7	INT_INTERRUPT_RETURN_ERR	R	0h	INT INTerrupt return error Reset type: PORESETn
6	INT_VECT_UNCERR	R	0h	INT Uncorrectable vector error Reset type: PORESETn
5	INT_VECT_CERR	R	0h	INT Correctable vector error Reset type: PORESETn
4	INT_ISR_ENTRY_ERR	R	0h	INT ISR entry error Reset type: PORESETn
3	MAIN_INTERRUPT_RETURN_ERR	R	0h	Main Interrupt return error Reset type: PORESETn
2	MAIN_VECT_UNCERR	R	0h	Main Uncorrectable vector error Reset type: PORESETn
1	MAIN_VECT_CERR	R	0h	Main Correctable vector error Reset type: PORESETn
0	MAIN_ISR_ENTRY_ERR	R	0h	Main ISR entry error Reset type: PORESETn

8.9.3.58 CPU2_INT_ERROR_TYPE_FRC Register (Offset = 24Ch) [Reset = 00000000h]

CPU2_INT_ERROR_TYPE_FRC is shown in Figure 8-66 and described in Table 8-83.

Return to the Summary Table.

CPU2 INT Error aggregator Error Type Force Register

Figure 8-66. CPU2_INT_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	PIPE_LOCK_KEY_ERR	REG_PARITY_DIAG_ERR	REG_PARITY_ERR	PIPE_SECURITY_VIO	MAXISP	WARNISP
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
VECT_UNCERR	VECT_CERR	NMI_CONTEXT_RESTORE_VECT_UNCERR	NMI_CONTEXT_RESTORE_VECT_CERR	NMI_INTERRUPT_RETURN_ERR	NMI_VECT_UNCERR	NMI_VECT_CERR	NMI_ISR_ENTRY_ERR
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
NMI_MAXISP_ERR	RESERVED	RTINT_CONTEXT_RESTORE_VECT_UNCERR	RTINT_CONTEXT_RESTORE_VECT_CERR	RTINT_INTERRUPT_RETURN_ERR	RTINT_VECT_UNCERR	RTINT_VECT_CERR	RTINT_ISR_ENTRY_ERR
R-0/W1S-0h	R-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
INT_INTERRUPT_RETURN_ERR	INT_VECT_UNCERR	INT_VECT_CERR	INT_ISR_ENTRY_ERR	MAIN_INTERRUPT_RETURN_ERR	MAIN_VECT_UNCERR	MAIN_VECT_CERR	MAIN_ISR_ENTRY_ERR
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 8-83. CPU2_INT_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29	PIPE_LOCK_KEY_ERR	R-0/W1S	0h	PIPE Lock key error force register 0 - No action 1 - force PIPE lock key error flag Reset type: PORESETn
28	REG_PARITY_DIAG_ERR	R-0/W1S	0h	PIPE Register parity diag error force register 0 - No action 1 - force MAX PSP violation flag Reset type: PORESETn
27	REG_PARITY_ERR	R-0/W1S	0h	PIPE Register parity error force register 0 - No action 1 - force PIPE Register parity errort error flag Reset type: PORESETn
26	PIPE_SECURITY_VIO	R-0/W1S	0h	PIPE security violation force register 0 - No action 1 - force PIPE security violation flag Reset type: PORESETn

Table 8-83. CPU2_INT_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	MAXISP	R-0/W1S	0h	Maxisp error flag force register 0 - No action 1 - force Maxisp error flag Reset type: PORESETn
24	WARNISP	R-0/W1S	0h	Warnisp error flag force register 0 - No action 1 - force Warnisp error flag Reset type: PORESETn
23	VECT_UNCERR	R-0/W1S	0h	Vector uncorrectable error force register 0 - No action 1 - force vector uncorrectable eror flag Reset type: PORESETn
22	VECT_CERR	R-0/W1S	0h	Vector correctable error force register 0 - No action 1 - force vector correctable error flag Reset type: PORESETn
21	NMI_CONTEXT_RESTORE_VECT_UNCERR	R-0/W1S	0h	NMI Context restore Uncorrectable vector error flag force register 0 - No action 1 - force NMI Context restore Uncorrectable vector error flag Reset type: PORESETn
20	NMI_CONTEXT_RESTORE_VECT_CERR	R-0/W1S	0h	NMI Context restore Correctable vector error flag force register 0 - No action 1 - force NMI Context restore Correctable vector error flag Reset type: PORESETn
19	NMI_INTERRUPT_RETURN_ERR	R-0/W1S	0h	NMI Interrupt return error flag force register 0 - No action 1 - force NMI Interrupt return error flag Reset type: PORESETn
18	NMI_VECT_UNCERR	R-0/W1S	0h	NMI uncorrectable vector error flag force register 0 - No action 1 - force NMI uncorrectable vector error flag Reset type: PORESETn
17	NMI_VECT_CERR	R-0/W1S	0h	NMI Correctable vector error flag force register 0 - No action 1 - force NMI Correctable vector error flag Reset type: PORESETn
16	NMI_ISR_ENTRY_ERR	R-0/W1S	0h	NMI ISR entry error flag force register 0 - No action 1 - NMI ISR entry error flag Reset type: PORESETn
15	NMI_MAXISP_ERR	R-0/W1S	0h	NMI MAXISP error flag force register 0 - No action 1 - NMI MAXISP error flag Reset type: PORESETn
14	RESERVED	R	0h	Reserved
13	RTINT_CONTEXT_RESTORE_VECT_UNCERR	R-0/W1S	0h	RTINT Context restore Uncorrectable vector error flag force register 0 - No action 1 - force RTINT Context restore Uncorrectable vector error flag Reset type: PORESETn
12	RTINT_CONTEXT_RESTORE_VECT_CERR	R-0/W1S	0h	RTINT Context restore Correctable vector error flag force register 0 - No action 1 - force RTINT Context restore Correctable vector error flag Reset type: PORESETn

Table 8-83. CPU2_INT_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	RTINT_INTERRUPT_RETURN_ERR	R-0/W1S	0h	RTINT Interrupt return error flag force register 0 - No action 1 - force RTINT Interrupt return error flag Reset type: PORESETn
10	RTINT_VECT_UNCERR	R-0/W1S	0h	RTINT uncorrectable vector error flag force register 0 - No action 1 - force RTINT uncorrectable vector error flag Reset type: PORESETn
9	RTINT_VECT_CERR	R-0/W1S	0h	RTINT Correctable vector error flag force register 0 - No action 1 - force RTINT Correctable vector error flag Reset type: PORESETn
8	RTINT_ISR_ENTRY_ERR	R-0/W1S	0h	RTINT ISR entry error flag force register 0 - No action 1 - RTINT ISR entry error flag Reset type: PORESETn
7	INT_INTERRUPT_RETURN_ERR	R-0/W1S	0h	INT Interrupt return error flag force register 0 - No action 1 - force INT Interrupt return error flag Reset type: PORESETn
6	INT_VECT_UNCERR	R-0/W1S	0h	INT uncorrectable vector error flag force register 0 - No action 1 - force INT uncorrectable vector error flag Reset type: PORESETn
5	INT_VECT_CERR	R-0/W1S	0h	INT Correctable vector error flag force register 0 - No action 1 - force INT Correctable vector error flag Reset type: PORESETn
4	INT_ISR_ENTRY_ERR	R-0/W1S	0h	INT ISR entry error flag force register 0 - No action 1 - INT ISR entry error flag Reset type: PORESETn
3	MAIN_INTERRUPT_RETURN_ERR	R-0/W1S	0h	Main Interrupt return error flag force register 0 - No action 1 - force Main Interrupt return error flag Reset type: PORESETn
2	MAIN_VECT_UNCERR	R-0/W1S	0h	Main uncorrectable vector error flag force register 0 - No action 1 - force Main uncorrectable vector error flag Reset type: PORESETn
1	MAIN_VECT_CERR	R-0/W1S	0h	Main Correctable vector error flag force register 0 - No action 1 - force Main Correctable vector error flag Reset type: PORESETn
0	MAIN_ISR_ENTRY_ERR	R-0/W1S	0h	Main ISR entry error flag force register 0 - No action 1 - Main ISR entry error flag Reset type: PORESETn

8.9.3.59 CPU2_INT_ERROR_TYPE_CLR Register (Offset = 250h) [Reset = 00000000h]

CPU2_INT_ERROR_TYPE_CLR is shown in Figure 8-67 and described in Table 8-84.

Return to the Summary Table.

CPU2 INT Error aggregator Error Type Clear Register

Figure 8-67. CPU2_INT_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	PIPE_LOCK_KEY_ERR	REG_PARITY_DIAG_ERR	REG_PARITY_ERR	PIPE_SECURITY_VIO	MAXISP	WARNISP
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
23	22	21	20	19	18	17	16
VECT_UNCERR	VECT_CERR	NMI_CONTEXT_RESTORE_VECT_UNCERR	NMI_CONTEXT_RESTORE_VECT_CERR	NMI_INTERRUPT_RETURN_ERR	NMI_VECT_UNCERR	NMI_VECT_CERR	NMI_ISR_ENTRY_ERR
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
15	14	13	12	11	10	9	8
NMI_MAXISP_ERR	RESERVED	RTINT_CONTEXT_RESTORE_VECT_UNCERR	RTINT_CONTEXT_RESTORE_VECT_CERR	RTINT_INTERRUPT_RETURN_ERR	RTINT_VECT_UNCERR	RTINT_VECT_CERR	RTINT_ISR_ENTRY_ERR
R-0/W1C-0h	R-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
INT_INTERRUPT_RETURN_ERR	INT_VECT_UNCERR	INT_VECT_CERR	INT_ISR_ENTRY_ERR	MAIN_INTERRUPT_RETURN_ERR	MAIN_VECT_UNCERR	MAIN_VECT_CERR	MAIN_ISR_ENTRY_ERR
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 8-84. CPU2_INT_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29	PIPE_LOCK_KEY_ERR	R-0/W1C	0h	PIPE Lock key error clear register 0 - No action 1 - clear PIPE lock key error flag Reset type: PORESETn
28	REG_PARITY_DIAG_ERR	R-0/W1C	0h	PIPE Register parity diag error clear register 0 - No action 1 - clear MAX PSP violation flag Reset type: PORESETn
27	REG_PARITY_ERR	R-0/W1C	0h	PIPE Register parity error clear register 0 - No action 1 - clear PIPE Register parity errort error flag Reset type: PORESETn
26	PIPE_SECURITY_VIO	R-0/W1C	0h	PIPE security violation clear register 0 - No action 1 - clear PIPE security violation flag Reset type: PORESETn

Table 8-84. CPU2_INT_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	MAXISP	R-0/W1C	0h	Maxisp error flag clear register 0 - No action 1 - Clear Maxisp error flag Reset type: PORESETn
24	WARNISP	R-0/W1C	0h	Warnisp error flag clear register 0 - No action 1 - Clear Warnisp error flag Reset type: PORESETn
23	VECT_UNCERR	R-0/W1C	0h	Vector uncorrectable error clear register 0 - No action 1 - Clear vector uncorrectable error flag Reset type: PORESETn
22	VECT_CERR	R-0/W1C	0h	Vector correctable error clear register 0 - No action 1 - Clear vector correctable error flag Reset type: PORESETn
21	NMI_CONTEXT_RESTORE_VECT_UNCERR	R-0/W1C	0h	NMI Context restore Uncorrectable vector error flag clear register 0 - No action 1 - clear NMI Context restore Uncorrectable vector error flag Reset type: PORESETn
20	NMI_CONTEXT_RESTORE_VECT_CERR	R-0/W1C	0h	NMI Context restore Correctable vector error flag clear register 0 - No action 1 - clear NMI Context restore Correctable vector error flag Reset type: PORESETn
19	NMI_INTERRUPT_RETURN_ERR	R-0/W1C	0h	NMI Interrupt return error flag clear register 0 - No action 1 - clear NMI Interrupt return error flag Reset type: PORESETn
18	NMI_VECT_UNCERR	R-0/W1C	0h	NMI uncorrectable vector error flag clear register 0 - No action 1 - clear NMI uncorrectable vector error flag Reset type: PORESETn
17	NMI_VECT_CERR	R-0/W1C	0h	NMI Correctable vector error flag clear register 0 - No action 1 - clear NMI Correctable vector error flag Reset type: PORESETn
16	NMI_ISR_ENTRY_ERR	R-0/W1C	0h	NMI ISR entry error flag clear register 0 - No action 1 - NMI ISR entry error flag Reset type: PORESETn
15	NMI_MAXISP_ERR	R-0/W1C	0h	NMI MAXISP error flag clear register 0 - No action 1 - NMI MAXISP error flag Reset type: PORESETn
14	RESERVED	R	0h	Reserved
13	RTINT_CONTEXT_RESTORE_VECT_UNCERR	R-0/W1C	0h	RTINT Context restore Uncorrectable vector error flag clear register 0 - No action 1 - clear RTINT Context restore Uncorrectable vector error flag Reset type: PORESETn
12	RTINT_CONTEXT_RESTORE_VECT_CERR	R-0/W1C	0h	RTINT Context restore Correctable vector error flag clear register 0 - No action 1 - clear RTINT Context restore Correctable vector error flag Reset type: PORESETn

Table 8-84. CPU2_INT_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	RTINT_INTERRUPT_RETURN_ERR	R-0/W1C	0h	RTINT Interrupt return error flag clear register 0 - No action 1 - clear RTINT Interrupt return error flag Reset type: PORESETn
10	RTINT_VECT_UNCERR	R-0/W1C	0h	RTINT uncorrectable vector error flag clear register 0 - No action 1 - clear RTINT uncorrectable vector error flag Reset type: PORESETn
9	RTINT_VECT_CERR	R-0/W1C	0h	RTINT Correctable vector error flag clear register 0 - No action 1 - clear RTINT Correctable vector error flag Reset type: PORESETn
8	RTINT_ISR_ENTRY_ERR	R-0/W1C	0h	RTINT ISR entry error flag clear register 0 - No action 1 - RTINT ISR entry error flag Reset type: PORESETn
7	INT_INTERRUPT_RETURN_ERR	R-0/W1C	0h	INT Interrupt return error flag clear register 0 - No action 1 - clear INT Interrupt return error flag Reset type: PORESETn
6	INT_VECT_UNCERR	R-0/W1C	0h	INT uncorrectable vector error flag clear register 0 - No action 1 - clear INT uncorrectable vector error flag Reset type: PORESETn
5	INT_VECT_CERR	R-0/W1C	0h	INT Correctable vector error flag clear register 0 - No action 1 - clear INT Correctable vector error flag Reset type: PORESETn
4	INT_ISR_ENTRY_ERR	R-0/W1C	0h	INT ISR entry error flag clear register 0 - No action 1 - INT ISR entry error flag Reset type: PORESETn
3	MAIN_INTERRUPT_RETURN_ERR	R-0/W1C	0h	Main Interrupt return error flag clear register 0 - No action 1 - clear Main Interrupt return error flag Reset type: PORESETn
2	MAIN_VECT_UNCERR	R-0/W1C	0h	Main uncorrectable vector error flag clear register 0 - No action 1 - clear Main uncorrectable vector error flag Reset type: PORESETn
1	MAIN_VECT_CERR	R-0/W1C	0h	Main Correctable vector error flag clear register 0 - No action 1 - clear Main Correctable vector error flag Reset type: PORESETn
0	MAIN_ISR_ENTRY_ERR	R-0/W1C	0h	Main ISR entry error flag clear register 0 - No action 1 - Main ISR entry error flag Reset type: PORESETn

8.9.3.60 CPU2_INT_PC Register (Offset = 254h) [Reset = 00000000h]

CPU2_INT_PC is shown in Figure 8-68 and described in Table 8-85.

Return to the Summary Table.

CPU2 INT Error aggregator Register to store PC value at the first High priority error event

Figure 8-68. CPU2_INT_PC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU2_INT_PC																															
R-0h																															

Table 8-85. CPU2_INT_PC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU2_INT_PC	R	0h	CPU2 PC at which first High priority error occurred on CPU2_INT Reset type: PORESETn

8.9.3.61 CPU3_PR_HIGHPRIO_ERROR_ADDRESS Register (Offset = 280h) [Reset = 00000000h]

CPU3_PR_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-69 and described in Table 8-86.

Return to the Summary Table.

CPU3 PR Error aggregator High Priority Error address register

Figure 8-69. CPU3_PR_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU3_PR_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-86. CPU3_PR_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU3_PR_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on CPU3_PR access Reset type: PORESETn

8.9.3.62 CPU3_PR_LOWPRIO_ERROR_ADDRESS Register (Offset = 284h) [Reset = 00000000h]

CPU3_PR_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-70 and described in Table 8-87.

Return to the Summary Table.

CPU3 PR Error aggregator Low Priority Error address register

Figure 8-70. CPU3_PR_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU3_PR_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-87. CPU3_PR_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU3_PR_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on CPU3_PR access Reset type: PORESETn

8.9.3.63 CPU3_PR_ERROR_TYPE Register (Offset = 288h) [Reset = 00000000h]

CPU3_PR_ERROR_TYPE is shown in Figure 8-71 and described in Table 8-88.

Return to the Summary Table.

CPU3 PR Error aggregator Error Type Register

Figure 8-71. CPU3_PR_ERROR_TYPE Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0h	R-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				INSTRUCTION_TIMEOUT	ILLEGAL_INSTRUCTION	SW_BREAKPOINT_ERR	WARN_PSP_ERR
R-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	MAX_PSP_ERR	SEC_EXIT_VIO	SEC_ENTRY_VIO	INSTR_SECURITY_VIO
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 8-88. CPU3_PR_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority set by software Reset type: PORESETn
29-12	RESERVED	R	0h	Reserved
11	INSTRUCTION_TIMEOUT	R	0h	Instruction timeout error Reset type: PORESETn
10	ILLEGAL_INSTRUCTION	R	0h	Illegal instruction error Reset type: PORESETn
9	SW_BREAKPOINT_ERR	R	0h	Software brakepoint error Reset type: PORESETn
8	WARN_PSP_ERR	R	0h	Warn PSP error Reset type: PORESETn
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3	MAX_PSP_ERR	R	0h	MAX PSP error Reset type: PORESETn
2	SEC_EXIT_VIO	R	0h	Secure exit error Reset type: PORESETn

Table 8-88. CPU3_PR_ERROR_TYPE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	SEC_ENTRY_VIO	R	0h	Secure entry and linear code crossing LINK, STACK, ZONE error. Reset type: PORESETn
0	INSTR_SECURITY_VIO	R	0h	Instruction packet security violation. Instruction packet crossed LINK, STACK, ZONE boundary. Reset type: PORESETn

8.9.3.64 CPU3_PR_ERROR_TYPE_FRC Register (Offset = 28Ch) [Reset = 00000000h]

CPU3_PR_ERROR_TYPE_FRC is shown in Figure 8-72 and described in Table 8-89.

Return to the Summary Table.

CPU3 PR Error aggregator Error Type Force Register

Figure 8-72. CPU3_PR_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				INSTRUCTION_TIMEOUT	ILLEGAL_INSTRUCTION	SW_BREAKPOINT_ERR	WARN_PSP_ERR
R-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	MAX_PSP_ERR	SEC_EXIT_VIO	SEC_ENTRY_VIO	INSTR_SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 8-89. CPU3_PR_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	Force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-12	RESERVED	R	0h	Reserved
11	INSTRUCTION_TIMEOUT	R-0/W1S	0h	Instruction timeout error flag force register 0 - No action 1 - force Instruction timeout flag Reset type: PORESETn
10	ILLEGAL_INSTRUCTION	R-0/W1S	0h	Illegal instruction error flag force register 0 - No action 1 - Illegal instruction error flag Reset type: PORESETn
9	SW_BREAKPOINT_ERR	R-0/W1S	0h	Software brakepoint error flag force register 0 - No action 1 - force Software brakepoint error flag Reset type: PORESETn
8	WARN_PSP_ERR	R-0/W1S	0h	WARN PSP violation flag force register 0 - No action 1 - force WARN PSP violation flag Reset type: PORESETn

Table 8-89. CPU3_PR_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3	MAX_PSP_ERR	R-0/W1S	0h	MAX PSP violation flag force register 0 - No action 1 - force MAX PSP violation flag Reset type: PORESETn
2	SEC_EXIT_VIO	R-0/W1S	0h	Secure exit violation flag force register 0 - No action 1 - force Secure exit violation flag Reset type: PORESETn
1	SEC_ENTRY_VIO	R-0/W1S	0h	Secure entry violation flag force register 0 - No action 1 - force Secure entry violation flag Reset type: PORESETn
0	INSTR_SECURITY_VIO	R-0/W1S	0h	Instruction packet security violation flag force register 0 - No action 1 - force Instruction packet security violation flag Reset type: PORESETn

8.9.3.65 CPU3_PR_ERROR_TYPE_CLR Register (Offset = 290h) [Reset = 00000000h]

CPU3_PR_ERROR_TYPE_CLR is shown in Figure 8-73 and described in Table 8-90.

Return to the Summary Table.

CPU3 PR Error aggregator Error Type Clear Register

Figure 8-73. CPU3_PR_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				INSTRUCTION_TIMEOUT	ILLEGAL_INSTRUCTION	SW_BREAKPOINT_ERR	WARN_PSP_ERR
R-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	MAX_PSP_ERR	SEC_EXIT_VIO	SEC_ENTRY_VIO	INSTR_SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 8-90. CPU3_PR_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-12	RESERVED	R	0h	Reserved
11	INSTRUCTION_TIMEOUT	R-0/W1C	0h	Instruction timeout error flag clear register 0 - No action 1 - clear Instruction timeout flag Reset type: PORESETn
10	ILLEGAL_INSTRUCTION	R-0/W1C	0h	Illegal instruction error flag clear register 0 - No action 1 - Illegal instruction error flag Reset type: PORESETn
9	SW_BREAKPOINT_ERR	R-0/W1C	0h	Software brakepoint error flag clear register 0 - No action 1 - clear Software brakepoint error flag Reset type: PORESETn
8	WARN_PSP_ERR	R-0/W1C	0h	WARN PSP violation flag clear register 0 - No action 1 - Clear WARN PSP violation flag Reset type: PORESETn

Table 8-90. CPU3_PR_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3	MAX_PSP_ERR	R-0/W1C	0h	MAX PSP violation flag clear register 0 - No action 1 - Clear MAX PSP violation flag Reset type: PORESETn
2	SEC_EXIT_VIO	R-0/W1C	0h	Secure exit violation flag clear register 0 - No action 1 - Clear Secure exit violation flag Reset type: PORESETn
1	SEC_ENTRY_VIO	R-0/W1C	0h	Secure entry violation flag clear register 0 - No action 1 - Clear Secure entry violation flag Reset type: PORESETn
0	INSTR_SECURITY_VIO	R-0/W1C	0h	Instruction packet security violation flag clear register 0 - No action 1 - Clear Instruction packet security violation flag Reset type: PORESETn

8.9.3.66 CPU3_PR_PC Register (Offset = 294h) [Reset = 00000000h]

CPU3_PR_PC is shown in Figure 8-74 and described in Table 8-91.

Return to the Summary Table.

CPU3 PR Error aggregator Register to store PC value at the first High priority error event

Figure 8-74. CPU3_PR_PC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU3_PR_PC																															
R-0h																															

Table 8-91. CPU3_PR_PC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU3_PR_PC	R	0h	CPPU3 PC at which first High priority error occurred on CPU3_PR access Reset type: PORESETn

8.9.3.67 CPU3_DR1_HIGHPRIO_ERROR_ADDRESS Register (Offset = 2C0h) [Reset = 00000000h]

CPU3_DR1_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-75 and described in Table 8-92.

Return to the Summary Table.

CPU3 DR1 Error aggregator High Priority Error address register

Figure 8-75. CPU3_DR1_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU3_DR1_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-92. CPU3_DR1_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU3_DR1_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on CPU3_DR1 access Reset type: PORESETn

8.9.3.68 CPU3_DR1_LOWPRIO_ERROR_ADDRESS Register (Offset = 2C4h) [Reset = 00000000h]

CPU3_DR1_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-76 and described in Table 8-93.

Return to the Summary Table.

CPU3 DR1 Error aggregator Low Priority Error address register

Figure 8-76. CPU3_DR1_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU3_DR1_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-93. CPU3_DR1_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU3_DR1_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on CPU3_DR1 access Reset type: PORESETn

8.9.3.69 CPU3_DR1_ERROR_TYPE Register (Offset = 2C8h) [Reset = 00000000h]

CPU3_DR1_ERROR_TYPE is shown in Figure 8-77 and described in Table 8-94.

Return to the Summary Table.

CPU3 DR1 Error aggregator Error Type Register

Figure 8-77. CPU3_DR1_ERROR_TYPE Register

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

Table 8-94. CPU3_DR1_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority high priority error set by software Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R	0h	Unaligned address error Reset type: PORESETn
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R	0h	Security violation Reset type: PORESETn

8.9.3.70 CPU3_DR1_ERROR_TYPE_FRC Register (Offset = 2CCh) [Reset = 00000000h]

CPU3_DR1_ERROR_TYPE_FRC is shown in Figure 8-78 and described in Table 8-95.

Return to the Summary Table.

CPU3 DR1 Error aggregator Error Type Force Register

Figure 8-78. CPU3_DR1_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1S-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h			R-0/W1S-0h

Table 8-95. CPU3_DR1_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1S	0h	Unaligned address error flag force register 0 - No action 1 - force Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn

Table 8-95. CPU3_DR1_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1S	0h	Security violation flag force register 0 - No action 1 - force Security violation flag Reset type: PORESETn

8.9.3.71 CPU3_DR1_ERROR_TYPE_CLR Register (Offset = 2D0h) [Reset = 00000000h]

CPU3_DR1_ERROR_TYPE_CLR is shown in Figure 8-79 and described in Table 8-96.

Return to the Summary Table.

CPU3 DR1 Error aggregator Error Type Clear Register

Figure 8-79. CPU3_DR1_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1C-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0h			R-0/W1C-0h

Table 8-96. CPU3_DR1_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1C	0h	Unaligned address error flag clear register 0 - No action 1 - Clear Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn

Table 8-96. CPU3_DR1_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1C	0h	Security violation flag clear register 0 - No action 1 - Clear Security violation flag Reset type: PORESETn

8.9.3.72 CPU3_DR1_PC Register (Offset = 2D4h) [Reset = 00000000h]

CPU3_DR1_PC is shown in Figure 8-80 and described in Table 8-97.

Return to the Summary Table.

CPU3 DR1 Error aggregator Register to store PC value at the first High priority error event

Figure 8-80. CPU3_DR1_PC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU3_DR1_PC																															
R-0h																															

Table 8-97. CPU3_DR1_PC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU3_DR1_PC	R	0h	CPU3 PC at which first High priority error occurred on CPU3_DR1 access Reset type: PORESETn

8.9.3.73 CPU3_DR2_HIGHPRIO_ERROR_ADDRESS Register (Offset = 300h) [Reset = 00000000h]

CPU3_DR2_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-81 and described in Table 8-98.

Return to the Summary Table.

CPU3 DR2 Error aggregator High Priority Error address register

Figure 8-81. CPU3_DR2_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU3_DR2_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-98. CPU3_DR2_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU3_DR2_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on CPU3_DR2 access Reset type: PORESETn

8.9.3.74 CPU3_DR2_LOWPRIO_ERROR_ADDRESS Register (Offset = 304h) [Reset = 00000000h]

CPU3_DR2_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-82 and described in Table 8-99.

Return to the Summary Table.

CPU3 DR2 Error aggregator Low Priority Error address register

Figure 8-82. CPU3_DR2_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU3_DR2_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-99. CPU3_DR2_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU3_DR2_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on CPU3_DR2 access Reset type: PORESETn

8.9.3.75 CPU3_DR2_ERROR_TYPE Register (Offset = 308h) [Reset = 00000000h]

CPU3_DR2_ERROR_TYPE is shown in Figure 8-83 and described in Table 8-100.

Return to the Summary Table.

CPU3 DR2 Error aggregator Error Type Register

Figure 8-83. CPU3_DR2_ERROR_TYPE Register

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

Table 8-100. CPU3_DR2_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority error set by software Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R	0h	Unaligned address error Reset type: PORESETn
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R	0h	Security violation Reset type: PORESETn

8.9.3.76 CPU3_DR2_ERROR_TYPE_FRC Register (Offset = 30Ch) [Reset = 00000000h]

CPU3_DR2_ERROR_TYPE_FRC is shown in Figure 8-84 and described in Table 8-101.

Return to the Summary Table.

CPU3 DR2 Error aggregator Error Type Force Register

Figure 8-84. CPU3_DR2_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1S-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h			R-0/W1S-0h

Table 8-101. CPU3_DR2_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1S	0h	Unaligned address error flag force register 0 - No action 1 - force Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn

Table 8-101. CPU3_DR2_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1S	0h	Security violation flag force register 0 - No action 1 - force Security violation flag Reset type: PORESETn

8.9.3.77 CPU3_DR2_ERROR_TYPE_CLR Register (Offset = 310h) [Reset = 00000000h]

CPU3_DR2_ERROR_TYPE_CLR is shown in Figure 8-85 and described in Table 8-102.

Return to the Summary Table.

CPU3 DR2 Error aggregator Error Type Clear Register

Figure 8-85. CPU3_DR2_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1C-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0h			R-0/W1C-0h

Table 8-102. CPU3_DR2_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1C	0h	Unaligned address error flag clear register 0 - No action 1 - Clear Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn

Table 8-102. CPU3_DR2_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1C	0h	Security violation flag clear register 0 - No action 1 - Clear Security violation flag Reset type: PORESETn

8.9.3.78 CPU3_DR2_PC Register (Offset = 314h) [Reset = 00000000h]

CPU3_DR2_PC is shown in Figure 8-86 and described in Table 8-103.

Return to the Summary Table.

CPU3 DR2 Error aggregator Register to store PC value at the first High priority error event

Figure 8-86. CPU3_DR2_PC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU3_DR2_PC																															
R-0h																															

Table 8-103. CPU3_DR2_PC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU3_DR2_PC	R	0h	CPU3 PC at which first High priority error occurred on CPU3_DR2 access Reset type: PORESETn

8.9.3.79 CPU3_DW_HIGHPRIO_ERROR_ADDRESS Register (Offset = 340h) [Reset = 00000000h]

CPU3_DW_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-87 and described in Table 8-104.

Return to the Summary Table.

CPU3 DW Error aggregator High Priority Error address register

Figure 8-87. CPU3_DW_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU3_DW_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-104. CPU3_DW_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU3_DW_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on CPU3_DW access Reset type: PORESETn

8.9.3.80 CPU3_DW_LOWPRIO_ERROR_ADDRESS Register (Offset = 344h) [Reset = 00000000h]

CPU3_DW_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-88 and described in Table 8-105.

Return to the Summary Table.

CPU3 DW Error aggregator Low Priority Error address register

Figure 8-88. CPU3_DW_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU3_DW_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-105. CPU3_DW_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU3_DW_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on CPU3_DW access Reset type: PORESETn

8.9.3.81 CPU3_DW_ERROR_TYPE Register (Offset = 348h) [Reset = 00000000h]

CPU3_DW_ERROR_TYPE is shown in Figure 8-89 and described in Table 8-106.

Return to the Summary Table.

CPU3 DW Error aggregator Error Type Register

Figure 8-89. CPU3_DW_ERROR_TYPE Register

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

Table 8-106. CPU3_DW_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority error set by software Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R	0h	Unaligned address error Reset type: PORESETn
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R	0h	Security violation Reset type: PORESETn

8.9.3.82 CPU3_DW_ERROR_TYPE_FRC Register (Offset = 34Ch) [Reset = 00000000h]

CPU3_DW_ERROR_TYPE_FRC is shown in Figure 8-90 and described in Table 8-107.

Return to the Summary Table.

CPU3 DW Error aggregator Error Type Force Register

Figure 8-90. CPU3_DW_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1S-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h			R-0/W1S-0h

Table 8-107. CPU3_DW_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1S	0h	Unaligned address error flag force register 0 - No action 1 - force Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn

Table 8-107. CPU3_DW_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1S	0h	Security violation flag force register 0 - No action 1 - force Security violation flag Reset type: PORESETn

8.9.3.83 CPU3_DW_ERROR_TYPE_CLR Register (Offset = 350h) [Reset = 00000000h]

CPU3_DW_ERROR_TYPE_CLR is shown in Figure 8-91 and described in Table 8-108.

Return to the Summary Table.

CPU3 DW Error aggregator Error Type Clear Register

Figure 8-91. CPU3_DW_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							UNALIGNED_ADDR_ERR
R-0h							R-0/W1C-0h
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0h			R-0/W1C-0h

Table 8-108. CPU3_DW_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-9	RESERVED	R	0h	Reserved
8	UNALIGNED_ADDR_ERR	R-0/W1C	0h	Unaligned address error flag clear register 0 - No action 1 - Clear Unaligned address error flag Reset type: PORESETn
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn

Table 8-108. CPU3_DW_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R-0/W1C	0h	Security violation flag clear register 0 - No action 1 - Clear Security violation flag Reset type: PORESETn

8.9.3.84 CPU3_DW_PC Register (Offset = 354h) [Reset = 00000000h]

CPU3_DW_PC is shown in Figure 8-92 and described in Table 8-109.

Return to the Summary Table.

CPU3 DW Error aggregator Register to store PC value at the first High priority error event

Figure 8-92. CPU3_DW_PC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU3_DW_PC																															
R-0h																															

Table 8-109. CPU3_DW_PC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU3_DW_PC	R	0h	CPU3 PC at which first High priority error occurred on CPU3_DW access Reset type: PORESETn

8.9.3.85 CPU3_INT_HIGHPRIO_ERROR_ADDRESS Register (Offset = 380h) [Reset = 00000000h]

CPU3_INT_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-93 and described in Table 8-110.

Return to the Summary Table.

CPU3 INT Error aggregator High Priority Error address register

Figure 8-93. CPU3_INT_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU3_INT_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-110. CPU3_INT_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU3_INT_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on CPU3_INT Reset type: PORESETn

8.9.3.86 CPU3_INT_LOWPRIO_ERROR_ADDRESS Register (Offset = 384h) [Reset = 00000000h]

CPU3_INT_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-94 and described in Table 8-111.

Return to the Summary Table.

CPU3 INT Error aggregator Low Priority Error address register

Figure 8-94. CPU3_INT_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU3_INT_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-111. CPU3_INT_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU3_INT_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on CPU3_INT Reset type: PORESETn

8.9.3.87 CPU3_INT_ERROR_TYPE Register (Offset = 388h) [Reset = 00000000h]

CPU3_INT_ERROR_TYPE is shown in Figure 8-95 and described in Table 8-112.

Return to the Summary Table.

CPU3 INT Error aggregator Error Type Register

Figure 8-95. CPU3_INT_ERROR_TYPE Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	PIPE_LOCK_KEY_ERR	REG_PARITY_DIAG_ERR	REG_PARITY_ERR	PIPE_SECURITY_VIO	MAXISP	WARNISP
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
VECT_UNCERR	VECT_CERR	NMI_CONTEXT_RESTORE_VECT_UNCERR	NMI_CONTEXT_RESTORE_VECT_CERR	NMI_INTERRUPT_RETURN_ERR	NMI_VECT_UNCERR	NMI_VECT_CERR	NMI_ISR_ENTRY_ERR
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
NMI_MAXISP_ERR	RESERVED	RTINT_CONTEXT_RESTORE_VECT_UNCERR	RTINT_CONTEXT_RESTORE_VECT_CERR	RTINT_INTERRUPT_RETURN_ERR	RTINT_VECT_UNCERR	RTINT_VECT_CERR	RTINT_ISR_ENTRY_ERR
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
INT_INTERRUPT_RETURN_ERR	INT_VECT_UNCERR	INT_VECT_CERR	INT_ISR_ENTRY_ERR	MAIN_INTERRUPT_RETURN_ERR	MAIN_VECT_UNCERR	MAIN_VECT_CERR	MAIN_ISR_ENTRY_ERR
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 8-112. CPU3_INT_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority error set by software Reset type: PORESETn
29	PIPE_LOCK_KEY_ERR	R	0h	PIPE Lock/key error Reset type: PORESETn
28	REG_PARITY_DIAG_ERR	R	0h	PIPE Register parity diag error Reset type: PORESETn
27	REG_PARITY_ERR	R	0h	PIPE Reg parity error Reset type: PORESETn
26	PIPE_SECURITY_VIO	R	0h	PIPE security violation Reset type: PORESETn
25	MAXISP	R	0h	maxisp Reset type: PORESETn
24	WARNISP	R	0h	warnisp Reset type: PORESETn
23	VECT_UNCERR	R	0h	Vector Uncorrectable error generated by PIPE on software read of vector register having uncorrectable error Reset type: PORESETn
22	VECT_CERR	R	0h	Vector Correctable error generated by PIPE on software read of vector register having correctable error Reset type: PORESETn

Table 8-112. CPU3_INT_ERROR_TYPE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	NMI_CONTEXT_RESTORE_VECT_UNCERR	R	0h	NMI Context restore Uncorrectable vector error Reset type: PORESETn
20	NMI_CONTEXT_RESTORE_VECT_CERR	R	0h	NMI Context restore Correctable vector error Reset type: PORESETn
19	NMI_INTERRUPT_RETURN_ERR	R	0h	NMI Interrupt return error Reset type: PORESETn
18	NMI_VECT_UNCERR	R	0h	NMI Uncorrectable vector error Reset type: PORESETn
17	NMI_VECT_CERR	R	0h	NMI Correctable vector error Reset type: PORESETn
16	NMI_ISR_ENTRY_ERR	R	0h	NMI ISR entry error Reset type: PORESETn
15	NMI_MAXISP_ERR	R	0h	NMI MAXISP error Reset type: PORESETn
14	RESERVED	R	0h	Reserved
13	RTINT_CONTEXT_RESTORE_VECT_UNCERR	R	0h	RTINT Context restore Uncorrectable vector error Reset type: PORESETn
12	RTINT_CONTEXT_RESTORE_VECT_CERR	R	0h	RTINT Context restore Correctable vector error Reset type: PORESETn
11	RTINT_INTERRUPT_RETURN_ERR	R	0h	RTINT Interrupt return error Reset type: PORESETn
10	RTINT_VECT_UNCERR	R	0h	RTINT Uncorrectable vector error Reset type: PORESETn
9	RTINT_VECT_CERR	R	0h	RTINT Correctable vector error Reset type: PORESETn
8	RTINT_ISR_ENTRY_ERR	R	0h	RTINT ISR entry error Reset type: PORESETn
7	INT_INTERRUPT_RETURN_ERR	R	0h	INT INTerrupt return error Reset type: PORESETn
6	INT_VECT_UNCERR	R	0h	INT Uncorrectable vector error Reset type: PORESETn
5	INT_VECT_CERR	R	0h	INT Correctable vector error Reset type: PORESETn
4	INT_ISR_ENTRY_ERR	R	0h	INT ISR entry error Reset type: PORESETn
3	MAIN_INTERRUPT_RETURN_ERR	R	0h	Main Interrupt return error Reset type: PORESETn
2	MAIN_VECT_UNCERR	R	0h	Main Uncorrectable vector error Reset type: PORESETn
1	MAIN_VECT_CERR	R	0h	Main Correctable vector error Reset type: PORESETn
0	MAIN_ISR_ENTRY_ERR	R	0h	Main ISR entry error Reset type: PORESETn

8.9.3.88 CPU3_INT_ERROR_TYPE_FRC Register (Offset = 38Ch) [Reset = 00000000h]

CPU3_INT_ERROR_TYPE_FRC is shown in Figure 8-96 and described in Table 8-113.

Return to the Summary Table.

CPU3 INT Error aggregator Error Type Force Register

Figure 8-96. CPU3_INT_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	PIPE_LOCK_KEY_ERR	REG_PARITY_DIAG_ERR	REG_PARITY_ERR	PIPE_SECURITY_VIO	MAXISP	WARNISP
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
VECT_UNCERR	VECT_CERR	NMI_CONTEXT_RESTORE_VECT_UNCERR	NMI_CONTEXT_RESTORE_VECT_CERR	NMI_INTERRUPT_RETURN_ERR	NMI_VECT_UNCERR	NMI_VECT_CERR	NMI_ISR_ENTRY_ERR
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
NMI_MAXISP_ERR	RESERVED	RTINT_CONTEXT_RESTORE_VECT_UNCERR	RTINT_CONTEXT_RESTORE_VECT_CERR	RTINT_INTERRUPT_RETURN_ERR	RTINT_VECT_UNCERR	RTINT_VECT_CERR	RTINT_ISR_ENTRY_ERR
R-0/W1S-0h	R-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
INT_INTERRUPT_RETURN_ERR	INT_VECT_UNCERR	INT_VECT_CERR	INT_ISR_ENTRY_ERR	MAIN_INTERRUPT_RETURN_ERR	MAIN_VECT_UNCERR	MAIN_VECT_CERR	MAIN_ISR_ENTRY_ERR
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 8-113. CPU3_INT_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29	PIPE_LOCK_KEY_ERR	R-0/W1S	0h	PIPE Lock key error force register 0 - No action 1 - force PIPE lock key error flag Reset type: PORESETn
28	REG_PARITY_DIAG_ERR	R-0/W1S	0h	PIPE Register parity diag error force register 0 - No action 1 - force MAX PSP violation flag Reset type: PORESETn
27	REG_PARITY_ERR	R-0/W1S	0h	PIPE Register parity error force register 0 - No action 1 - force PIPE Register parity errort error flag Reset type: PORESETn
26	PIPE_SECURITY_VIO	R-0/W1S	0h	PIPE security violation force register 0 - No action 1 - force PIPE security violation flag Reset type: PORESETn

Table 8-113. CPU3_INT_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	MAXISP	R-0/W1S	0h	Maxisp error flag force register 0 - No action 1 - force Maxisp error flag Reset type: PORESETn
24	WARNISP	R-0/W1S	0h	Warnisp error flag force register 0 - No action 1 - force Warnisp error flag Reset type: PORESETn
23	VECT_UNCERR	R-0/W1S	0h	Vector uncorrectable error force register 0 - No action 1 - force vector uncorrectable eror flag Reset type: PORESETn
22	VECT_CERR	R-0/W1S	0h	Vector correctable error force register 0 - No action 1 - force vector correctable error flag Reset type: PORESETn
21	NMI_CONTEXT_RESTORE_VECT_UNCERR	R-0/W1S	0h	NMI Context restore Uncorrectable vector error flag force register 0 - No action 1 - force NMI Context restore Uncorrectable vector error flag Reset type: PORESETn
20	NMI_CONTEXT_RESTORE_VECT_CERR	R-0/W1S	0h	NMI Context restore Correctable vector error flag force register 0 - No action 1 - force NMI Context restore Correctable vector error flag Reset type: PORESETn
19	NMI_INTERRUPT_RETURN_ERR	R-0/W1S	0h	NMI Interrupt return error flag force register 0 - No action 1 - force NMI Interrupt return error flag Reset type: PORESETn
18	NMI_VECT_UNCERR	R-0/W1S	0h	NMI uncorrectable vector error flag force register 0 - No action 1 - force NMI uncorrectable vector error flag Reset type: PORESETn
17	NMI_VECT_CERR	R-0/W1S	0h	NMI Correctable vector error flag force register 0 - No action 1 - force NMI Correctable vector error flag Reset type: PORESETn
16	NMI_ISR_ENTRY_ERR	R-0/W1S	0h	NMI ISR entry error flag force register 0 - No action 1 - NMI ISR entry error flag Reset type: PORESETn
15	NMI_MAXISP_ERR	R-0/W1S	0h	NMI MAXISP error flag force register 0 - No action 1 - NMI MAXISP error flag Reset type: PORESETn
14	RESERVED	R	0h	Reserved
13	RTINT_CONTEXT_RESTORE_VECT_UNCERR	R-0/W1S	0h	RTINT Context restore Uncorrectable vector error flag force register 0 - No action 1 - force RTINT Context restore Uncorrectable vector error flag Reset type: PORESETn
12	RTINT_CONTEXT_RESTORE_VECT_CERR	R-0/W1S	0h	RTINT Context restore Correctable vector error flag force register 0 - No action 1 - force RTINT Context restore Correctable vector error flag Reset type: PORESETn

Table 8-113. CPU3_INT_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	RTINT_INTERRUPT_RETURN_ERR	R-0/W1S	0h	RTINT Interrupt return error flag force register 0 - No action 1 - force RTINT Interrupt return error flag Reset type: PORESETn
10	RTINT_VECT_UNCERR	R-0/W1S	0h	RTINT uncorrectable vector error flag force register 0 - No action 1 - force RTINT uncorrectable vector error flag Reset type: PORESETn
9	RTINT_VECT_CERR	R-0/W1S	0h	RTINT Correctable vector error flag force register 0 - No action 1 - force RTINT Correctable vector error flag Reset type: PORESETn
8	RTINT_ISR_ENTRY_ERR	R-0/W1S	0h	RTINT ISR entry error flag force register 0 - No action 1 - RTINT ISR entry error flag Reset type: PORESETn
7	INT_INTERRUPT_RETURN_ERR	R-0/W1S	0h	INT Interrupt return error flag force register 0 - No action 1 - force INT Interrupt return error flag Reset type: PORESETn
6	INT_VECT_UNCERR	R-0/W1S	0h	INT uncorrectable vector error flag force register 0 - No action 1 - force INT uncorrectable vector error flag Reset type: PORESETn
5	INT_VECT_CERR	R-0/W1S	0h	INT Correctable vector error flag force register 0 - No action 1 - force INT Correctable vector error flag Reset type: PORESETn
4	INT_ISR_ENTRY_ERR	R-0/W1S	0h	INT ISR entry error flag force register 0 - No action 1 - INT ISR entry error flag Reset type: PORESETn
3	MAIN_INTERRUPT_RETURN_ERR	R-0/W1S	0h	Main Interrupt return error flag force register 0 - No action 1 - force Main Interrupt return error flag Reset type: PORESETn
2	MAIN_VECT_UNCERR	R-0/W1S	0h	Main uncorrectable vector error flag force register 0 - No action 1 - force Main uncorrectable vector error flag Reset type: PORESETn
1	MAIN_VECT_CERR	R-0/W1S	0h	Main Correctable vector error flag force register 0 - No action 1 - force Main Correctable vector error flag Reset type: PORESETn
0	MAIN_ISR_ENTRY_ERR	R-0/W1S	0h	Main ISR entry error flag force register 0 - No action 1 - Main ISR entry error flag Reset type: PORESETn

8.9.3.89 CPU3_INT_ERROR_TYPE_CLR Register (Offset = 390h) [Reset = 00000000h]

CPU3_INT_ERROR_TYPE_CLR is shown in Figure 8-97 and described in Table 8-114.

Return to the Summary Table.

CPU3 INT Error aggregator Error Type Clear Register

Figure 8-97. CPU3_INT_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	PIPE_LOCK_KEY_ERR	REG_PARITY_DIAG_ERR	REG_PARITY_ERR	PIPE_SECURITY_VIO	MAXISP	WARNISP
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
23	22	21	20	19	18	17	16
VECT_UNCERR	VECT_CERR	NMI_CONTEXT_RESTORE_VECT_UNCERR	NMI_CONTEXT_RESTORE_VECT_CERR	NMI_INTERRUPT_RETURN_ERR	NMI_VECT_UNCERR	NMI_VECT_CERR	NMI_ISR_ENTRY_ERR
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
15	14	13	12	11	10	9	8
NMI_MAXISP_ERR	RESERVED	RTINT_CONTEXT_RESTORE_VECT_UNCERR	RTINT_CONTEXT_RESTORE_VECT_CERR	RTINT_INTERRUPT_RETURN_ERR	RTINT_VECT_UNCERR	RTINT_VECT_CERR	RTINT_ISR_ENTRY_ERR
R-0/W1C-0h	R-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
INT_INTERRUPT_RETURN_ERR	INT_VECT_UNCERR	INT_VECT_CERR	INT_ISR_ENTRY_ERR	MAIN_INTERRUPT_RETURN_ERR	MAIN_VECT_UNCERR	MAIN_VECT_CERR	MAIN_ISR_ENTRY_ERR
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 8-114. CPU3_INT_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29	PIPE_LOCK_KEY_ERR	R-0/W1C	0h	PIPE Lock key error clear register 0 - No action 1 - clear PIPE lock key error flag Reset type: PORESETn
28	REG_PARITY_DIAG_ERR	R-0/W1C	0h	PIPE Register parity diag error clear register 0 - No action 1 - clear MAX PSP violation flag Reset type: PORESETn
27	REG_PARITY_ERR	R-0/W1C	0h	PIPE Register parity error clear register 0 - No action 1 - clear PIPE Register parity errort error flag Reset type: PORESETn
26	PIPE_SECURITY_VIO	R-0/W1C	0h	PIPE security violation clear register 0 - No action 1 - clear PIPE security violation flag Reset type: PORESETn

Table 8-114. CPU3_INT_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	MAXISP	R-0/W1C	0h	Maxisp error flag clear register 0 - No action 1 - Clear Maxisp error flag Reset type: PORESETn
24	WARNISP	R-0/W1C	0h	Warnisp error flag clear register 0 - No action 1 - Clear Warnisp error flag Reset type: PORESETn
23	VECT_UNCERR	R-0/W1C	0h	Vector uncorrectable error clear register 0 - No action 1 - Clear vector uncorrectable error flag Reset type: PORESETn
22	VECT_CERR	R-0/W1C	0h	Vector correctable error clear register 0 - No action 1 - Clear vector correctable error flag Reset type: PORESETn
21	NMI_CONTEXT_RESTORE_VECT_UNCERR	R-0/W1C	0h	NMI Context restore Uncorrectable vector error flag clear register 0 - No action 1 - clear NMI Context restore Uncorrectable vector error flag Reset type: PORESETn
20	NMI_CONTEXT_RESTORE_VECT_CERR	R-0/W1C	0h	NMI Context restore Correctable vector error flag clear register 0 - No action 1 - clear NMI Context restore Correctable vector error flag Reset type: PORESETn
19	NMI_INTERRUPT_RETURN_ERR	R-0/W1C	0h	NMI Interrupt return error flag clear register 0 - No action 1 - clear NMI Interrupt return error flag Reset type: PORESETn
18	NMI_VECT_UNCERR	R-0/W1C	0h	NMI uncorrectable vector error flag clear register 0 - No action 1 - clear NMI uncorrectable vector error flag Reset type: PORESETn
17	NMI_VECT_CERR	R-0/W1C	0h	NMI Correctable vector error flag clear register 0 - No action 1 - clear NMI Correctable vector error flag Reset type: PORESETn
16	NMI_ISR_ENTRY_ERR	R-0/W1C	0h	NMI ISR entry error flag clear register 0 - No action 1 - NMI ISR entry error flag Reset type: PORESETn
15	NMI_MAXISP_ERR	R-0/W1C	0h	NMI MAXISP error flag clear register 0 - No action 1 - NMI MAXISP error flag Reset type: PORESETn
14	RESERVED	R	0h	Reserved
13	RTINT_CONTEXT_RESTORE_VECT_UNCERR	R-0/W1C	0h	RTINT Context restore Uncorrectable vector error flag clear register 0 - No action 1 - clear RTINT Context restore Uncorrectable vector error flag Reset type: PORESETn
12	RTINT_CONTEXT_RESTORE_VECT_CERR	R-0/W1C	0h	RTINT Context restore Correctable vector error flag clear register 0 - No action 1 - clear RTINT Context restore Correctable vector error flag Reset type: PORESETn

Table 8-114. CPU3_INT_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	RTINT_INTERRUPT_RETURN_ERR	R-0/W1C	0h	RTINT Interrupt return error flag clear register 0 - No action 1 - clear RTINT Interrupt return error flag Reset type: PORESETn
10	RTINT_VECT_UNCERR	R-0/W1C	0h	RTINT uncorrectable vector error flag clear register 0 - No action 1 - clear RTINT uncorrectable vector error flag Reset type: PORESETn
9	RTINT_VECT_CERR	R-0/W1C	0h	RTINT Correctable vector error flag clear register 0 - No action 1 - clear RTINT Correctable vector error flag Reset type: PORESETn
8	RTINT_ISR_ENTRY_ERR	R-0/W1C	0h	RTINT ISR entry error flag clear register 0 - No action 1 - RTINT ISR entry error flag Reset type: PORESETn
7	INT_INTERRUPT_RETURN_ERR	R-0/W1C	0h	INT Interrupt return error flag clear register 0 - No action 1 - clear INT Interrupt return error flag Reset type: PORESETn
6	INT_VECT_UNCERR	R-0/W1C	0h	INT uncorrectable vector error flag clear register 0 - No action 1 - clear INT uncorrectable vector error flag Reset type: PORESETn
5	INT_VECT_CERR	R-0/W1C	0h	INT Correctable vector error flag clear register 0 - No action 1 - clear INT Correctable vector error flag Reset type: PORESETn
4	INT_ISR_ENTRY_ERR	R-0/W1C	0h	INT ISR entry error flag clear register 0 - No action 1 - INT ISR entry error flag Reset type: PORESETn
3	MAIN_INTERRUPT_RETURN_ERR	R-0/W1C	0h	Main Interrupt return error flag clear register 0 - No action 1 - clear Main Interrupt return error flag Reset type: PORESETn
2	MAIN_VECT_UNCERR	R-0/W1C	0h	Main uncorrectable vector error flag clear register 0 - No action 1 - clear Main uncorrectable vector error flag Reset type: PORESETn
1	MAIN_VECT_CERR	R-0/W1C	0h	Main Correctable vector error flag clear register 0 - No action 1 - clear Main Correctable vector error flag Reset type: PORESETn
0	MAIN_ISR_ENTRY_ERR	R-0/W1C	0h	Main ISR entry error flag clear register 0 - No action 1 - Main ISR entry error flag Reset type: PORESETn

8.9.3.90 CPU3_INT_PC Register (Offset = 394h) [Reset = 00000000h]

CPU3_INT_PC is shown in Figure 8-98 and described in Table 8-115.

Return to the Summary Table.

CPU3 INT Error aggregator Register to store PC value at the first High priority error event

Figure 8-98. CPU3_INT_PC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU3_INT_PC																															
R-0h																															

Table 8-115. CPU3_INT_PC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPU3_INT_PC	R	0h	CPU3 PC at which first High priority error occurred on CPU3_INT Reset type: PORESETn

8.9.3.91 RTDMA1_DR_HIGHPRIO_ERROR_ADDRESS Register (Offset = 780h) [Reset = 00000000h]

RTDMA1_DR_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-99 and described in Table 8-116.

Return to the Summary Table.

RTDMA1 DR Error aggregator High Priority Error address register

Figure 8-99. RTDMA1_DR_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTDMA1_DR_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-116. RTDMA1_DR_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTDMA1_DR_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on RTDMA1_DR access Reset type: PORESETn

8.9.3.92 RTDMA1_DR_LOWPRIO_ERROR_ADDRESS Register (Offset = 784h) [Reset = 00000000h]

RTDMA1_DR_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-100 and described in Table 8-117.

Return to the Summary Table.

RTDMA1 DR Error aggregator Low Priority Error address register

Figure 8-100. RTDMA1_DR_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTDMA1_DR_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-117. RTDMA1_DR_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTDMA1_DR_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on RTDMA1_DR access Reset type: PORESETn

8.9.3.93 RTDMA1_DR_ERROR_TYPE Register (Offset = 788h) [Reset = 00000000h]

RTDMA1_DR_ERROR_TYPE is shown in Figure 8-101 and described in Table 8-118.

Return to the Summary Table.

RTDMA1 DR Error aggregator Error Type Register

Figure 8-101. RTDMA1_DR_ERROR_TYPE Register

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

Table 8-118. RTDMA1_DR_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	low priority software error set by software Reset type: PORESETn
29-8	RESERVED	R	0h	Reserved
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R	0h	Security violation Reset type: PORESETn

8.9.3.94 RTDMA1_DR_ERROR_TYPE_FRC Register (Offset = 78Ch) [Reset = 00000000h]

RTDMA1_DR_ERROR_TYPE_FRC is shown in Figure 8-102 and described in Table 8-119.

Return to the Summary Table.

RTDMA1 DR Error aggregator Error Type Force Register

Figure 8-102. RTDMA1_DR_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h			R-0/W1S-0h

Table 8-119. RTDMA1_DR_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-8	RESERVED	R	0h	Reserved
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved

Table 8-119. RTDMA1_DR_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SECURITY_VIO	R-0/W1S	0h	Security violation flag force register 0 - No action 1 - force Security violation flag Reset type: PORESETn

8.9.3.95 RTDMA1_DR_ERROR_TYPE_CLR Register (Offset = 790h) [Reset = 00000000h]

RTDMA1_DR_ERROR_TYPE_CLR is shown in Figure 8-103 and described in Table 8-120.

Return to the Summary Table.

RTDMA1 DR Error aggregator Error Type Clear Register

Figure 8-103. RTDMA1_DR_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0h			R-0/W1C-0h

Table 8-120. RTDMA1_DR_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-8	RESERVED	R	0h	Reserved
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved

Table 8-120. RTDMA1_DR_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SECURITY_VIO	R-0/W1C	0h	Security violation flag clear register 0 - No action 1 - Clear Security violation flag Reset type: PORESETn

8.9.3.96 RTDMA1_DW_HIGHPRIO_ERROR_ADDRESS Register (Offset = 7C0h) [Reset = 00000000h]

RTDMA1_DW_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-104 and described in Table 8-121.

Return to the Summary Table.

RTDMA1 DW Error aggregator High Priority Error address register

Figure 8-104. RTDMA1_DW_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTDMA1_DW_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-121. RTDMA1_DW_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTDMA1_DW_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on RTDMA1_DW access Reset type: PORESETn

8.9.3.97 RTDMA1_DW_LOWPRIO_ERROR_ADDRESS Register (Offset = 7C4h) [Reset = 00000000h]

RTDMA1_DW_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-105 and described in Table 8-122.

Return to the Summary Table.

RTDMA1 DW Error aggregator Low Priority Error address register

Figure 8-105. RTDMA1_DW_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTDMA1_DW_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-122. RTDMA1_DW_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTDMA1_DW_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on RTDMA1_DW access Reset type: PORESETn

8.9.3.98 RTDMA1_DW_ERROR_TYPE Register (Offset = 7C8h) [Reset = 00000000h]

RTDMA1_DW_ERROR_TYPE is shown in Figure 8-106 and described in Table 8-123.

Return to the Summary Table.

RTDMA1 DW Error aggregator Error Type Register

Figure 8-106. RTDMA1_DW_ERROR_TYPE Register

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

Table 8-123. RTDMA1_DW_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority error set by software Reset type: PORESETn
29-8	RESERVED	R	0h	Reserved
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R	0h	Security violation Reset type: PORESETn

8.9.3.99 RTDMA1_DW_ERROR_TYPE_FRC Register (Offset = 7CCh) [Reset = 00000000h]

RTDMA1_DW_ERROR_TYPE_FRC is shown in Figure 8-107 and described in Table 8-124.

Return to the Summary Table.

RTDMA1 DW Error aggregator Error Type Force Register

Figure 8-107. RTDMA1_DW_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h			R-0/W1S-0h

Table 8-124. RTDMA1_DW_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-8	RESERVED	R	0h	Reserved
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved

Table 8-124. RTDMA1_DW_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SECURITY_VIO	R-0/W1S	0h	Security violation flag force register 0 - No action 1 - force Security violation flag Reset type: PORESETn

8.9.3.100 RTDMA1_DW_ERROR_TYPE_CLR Register (Offset = 7D0h) [Reset = 00000000h]

RTDMA1_DW_ERROR_TYPE_CLR is shown in Figure 8-108 and described in Table 8-125.

Return to the Summary Table.

RTDMA1 DW Error aggregator Error Type Clear Register

Figure 8-108. RTDMA1_DW_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0h			R-0/W1C-0h

Table 8-125. RTDMA1_DW_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-8	RESERVED	R	0h	Reserved
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved

Table 8-125. RTDMA1_DW_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SECURITY_VIO	R-0/W1C	0h	Security violation flag clear register 0 - No action 1 - Clear Security violation flag Reset type: PORESETn

8.9.3.101 RTDMA2_DR_HIGHPRIO_ERROR_ADDRESS Register (Offset = 800h) [Reset = 00000000h]

RTDMA2_DR_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-109 and described in Table 8-126.

Return to the Summary Table.

RTDMA2 DR Error aggregator High Priority Error address register

Figure 8-109. RTDMA2_DR_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTDMA2_DR_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-126. RTDMA2_DR_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTDMA2_DR_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on RTDMA2_DR access Reset type: PORESETn

8.9.3.102 RTDMA2_DR_LOWPRIO_ERROR_ADDRESS Register (Offset = 804h) [Reset = 00000000h]

RTDMA2_DR_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-110 and described in Table 8-127.

Return to the Summary Table.

RTDMA2 DR Error aggregator Low Priority Error address register

Figure 8-110. RTDMA2_DR_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTDMA2_DR_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-127. RTDMA2_DR_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTDMA2_DR_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on RTDMA2_DR access Reset type: PORESETn

8.9.3.103 RTDMA2_DR_ERROR_TYPE Register (Offset = 808h) [Reset = 00000000h]

RTDMA2_DR_ERROR_TYPE is shown in Figure 8-111 and described in Table 8-128.

Return to the Summary Table.

RTDMA2 DR Error aggregator Error Type Register

Figure 8-111. RTDMA2_DR_ERROR_TYPE Register

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

Table 8-128. RTDMA2_DR_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	low priority software error set by software Reset type: PORESETn
29-8	RESERVED	R	0h	Reserved
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R	0h	Security violation Reset type: PORESETn

8.9.3.104 RTDMA2_DR_ERROR_TYPE_FRC Register (Offset = 80Ch) [Reset = 00000000h]

RTDMA2_DR_ERROR_TYPE_FRC is shown in Figure 8-112 and described in Table 8-129.

Return to the Summary Table.

RTDMA2 DR Error aggregator Error Type Force Register

Figure 8-112. RTDMA2_DR_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h			R-0/W1S-0h

Table 8-129. RTDMA2_DR_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-8	RESERVED	R	0h	Reserved
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved

Table 8-129. RTDMA2_DR_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SECURITY_VIO	R-0/W1S	0h	Security violation flag force register 0 - No action 1 - force Security violation flag Reset type: PORESETn

8.9.3.105 RTDMA2_DR_ERROR_TYPE_CLR Register (Offset = 810h) [Reset = 00000000h]

RTDMA2_DR_ERROR_TYPE_CLR is shown in Figure 8-113 and described in Table 8-130.

Return to the Summary Table.

RTDMA2 DR Error aggregator Error Type Clear Register

Figure 8-113. RTDMA2_DR_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0h			R-0/W1C-0h

Table 8-130. RTDMA2_DR_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-8	RESERVED	R	0h	Reserved
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved

Table 8-130. RTDMA2_DR_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SECURITY_VIO	R-0/W1C	0h	Security violation flag clear register 0 - No action 1 - Clear Security violation flag Reset type: PORESETn

8.9.3.106 RTDMA2_DW_HIGHPRIO_ERROR_ADDRESS Register (Offset = 840h) [Reset = 00000000h]

RTDMA2_DW_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-114 and described in Table 8-131.

Return to the Summary Table.

RTDMA2 DW Error aggregator High Priority Error address register

Figure 8-114. RTDMA2_DW_HIGHPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTDMA2_DW_HIGHPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-131. RTDMA2_DW_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTDMA2_DW_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on RTDMA2_DW access Reset type: PORESETn

8.9.3.107 RTDMA2_DW_LOWPRIO_ERROR_ADDRESS Register (Offset = 844h) [Reset = 00000000h]

RTDMA2_DW_LOWPRIO_ERROR_ADDRESS is shown in Figure 8-115 and described in Table 8-132.

Return to the Summary Table.

RTDMA2 DW Error aggregator Low Priority Error address register

Figure 8-115. RTDMA2_DW_LOWPRIO_ERROR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTDMA2_DW_LOWPRIO_ERROR_ADDRESS																															
R-0h																															

Table 8-132. RTDMA2_DW_LOWPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTDMA2_DW_LOWPRIO_ERROR_ADDRESS	R	0h	Address at which first low priority error occurred on RTDMA2_DW access Reset type: PORESETn

8.9.3.108 RTDMA2_DW_ERROR_TYPE Register (Offset = 848h) [Reset = 00000000h]

RTDMA2_DW_ERROR_TYPE is shown in Figure 8-116 and described in Table 8-133.

Return to the Summary Table.

RTDMA2 DW Error aggregator Error Type Register

Figure 8-116. RTDMA2_DW_ERROR_TYPE Register

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

Table 8-133. RTDMA2_DW_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	LP_SOFTWARE_ERR	R	0h	Low priority error set by software Reset type: PORESETn
29-8	RESERVED	R	0h	Reserved
7	CERR	R	0h	Correctable error Reset type: PORESETn
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5	ACC_ACK_ERR	R	0h	Access ACK error Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R	0h	Access timeout error Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved
0	SECURITY_VIO	R	0h	Security violation Reset type: PORESETn

8.9.3.109 RTDMA2_DW_ERROR_TYPE_FRC Register (Offset = 84Ch) [Reset = 00000000h]

RTDMA2_DW_ERROR_TYPE_FRC is shown in Figure 8-117 and described in Table 8-134.

Return to the Summary Table.

RTDMA2 DW Error aggregator Error Type Force Register

Figure 8-117. RTDMA2_DW_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h			R-0/W1S-0h

Table 8-134. RTDMA2_DW_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1S	0h	Force low priority software error 0 - No action 1 - force low priority software error Reset type: PORESETn
29-8	RESERVED	R	0h	Reserved
7	CERR	R-0/W1S	0h	Correctable error flag force register 0 - No action 1 - force Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1S	0h	Access ACK error flag force register 0 - No action 1 - force Access ACK error flag Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R-0/W1S	0h	Access timeout error flag force register 0 - No action 1 - force Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved

Table 8-134. RTDMA2_DW_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SECURITY_VIO	R-0/W1S	0h	Security violation flag force register 0 - No action 1 - force Security violation flag Reset type: PORESETn

8.9.3.110 RTDMA2_DW_ERROR_TYPE_CLR Register (Offset = 850h) [Reset = 00000000h]

RTDMA2_DW_ERROR_TYPE_CLR is shown in Figure 8-118 and described in Table 8-135.

Return to the Summary Table.

RTDMA2 DW Error aggregator Error Type Clear Register

Figure 8-118. RTDMA2_DW_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	LP_SOFTWARE_ERR	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CERR	UNCERR	ACC_ACK_ERR	ACC_TIMEOUT_ERR	RESERVED			SECURITY_VIO
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0h			R-0/W1C-0h

Table 8-135. RTDMA2_DW_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	LP_SOFTWARE_ERR	R-0/W1C	0h	clear low priority software error 0 - No action 1 - clear low priority software error Reset type: PORESETn
29-8	RESERVED	R	0h	Reserved
7	CERR	R-0/W1C	0h	Correctable error flag clear register 0 - No action 1 - Clear Correctable error flag Reset type: PORESETn
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5	ACC_ACK_ERR	R-0/W1C	0h	Access ACK error flag clear register 0 - No action 1 - Clear Access ACK error flag Reset type: PORESETn
4	ACC_TIMEOUT_ERR	R-0/W1C	0h	Access timeout error flag clear register 0 - No action 1 - Clear Access timeout error flag Reset type: PORESETn
3-1	RESERVED	R	0h	Reserved

Table 8-135. RTDMA2_DW_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SECURITY_VIO	R-0/W1C	0h	Security violation flag clear register 0 - No action 1 - Clear Security violation flag Reset type: PORESETn

8.9.3.111 SSU_HIGHPRIO_ERROR_ADDRESS Register (Offset = 880h) [Reset = 00000000h]

SSU_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-119 and described in Table 8-136.

Return to the Summary Table.

SSU Error aggregator High Priority Error address register

Figure 8-119. SSU_HIGHPRIO_ERROR_ADDRESS Register

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

Table 8-136. SSU_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SSU_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first high priority SSU error occurred Reset type: PORESETn

8.9.3.112 SSU_ERROR_TYPE Register (Offset = 888h) [Reset = 00000000h]

SSU_ERROR_TYPE is shown in Figure 8-120 and described in Table 8-137.

Return to the Summary Table.

SSU Error aggregator Error Type Register

Figure 8-120. SSU_ERROR_TYPE Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	RESERVED	RESERVED					
R-0h	R-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED						FLC2_MMR_ACCESS_ERROR	FLC2_ERROR_TYPE
R-0h						R-0h	R-0h
15	14	13	12	11	10	9	8
FLC2_ERROR_TYPE		FLC2_ERROR_STS	FLC1_MMR_ACCESS_ERROR	FLC1_ERROR_TYPE			FLC1_ERROR_STS
R-0h		R-0h	R-0h	R-0h			R-0h
7	6	5	4	3	2	1	0
MODE_INVALID	RESERVED			CPU3_SSU_MMR_ACCESS_ERROR	CPU2_SSU_MMR_ACCESS_ERROR	CPU1_SSU_MMR_ACCESS_ERROR	RESERVED
R-0h	R-0h			R-0h	R-0h	R-0h	R-0h

Table 8-137. SSU_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	RESERVED	R	0h	Reserved
29-18	RESERVED	R	0h	Reserved
17	FLC2_MMR_ACCESS_ERROR	R	0h	FLC2_MMR_ACCESS_ERROR Reset type: PORESETn
16-14	FLC2_ERROR_TYPE	R	0h	FLC2_ERROR type 0x0: FLC2_ILLADDR 0x1: FLC2_ILLPROG 0x2: FLC2_ILLERASE 0x3: FLC2_ILLRDVER 0x4: FLC2_ILLMODECH 0x5: FLC2_ILLCMD 0x6: FLC2_ILLSIZE 0x7: FLC2_ILLBANKERASE Note: SSU_ERROR_TYPE.FLC2_ERROR_TYPE register will not get updated on further FLC2 errors until the SSU_ERROR_TYPE.FLC2_ERROR_TYPE bits are cleared. Reset type: PORESETn
13	FLC2_ERROR_STS	R	0h	FLC2_ERROR status Reset type: PORESETn
12	FLC1_MMR_ACCESS_ERROR	R	0h	FLC1_MMR_ACCESS_ERROR Reset type: PORESETn

Table 8-137. SSU_ERROR_TYPE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-9	FLC1_ERROR_TYPE	R	0h	FLC1_ERROR type register 0x0: FLC1_ILLADDR 0x1: FLC1_ILLPROG 0x2: FLC1_ILLERASE 0x3: FLC1_ILLRDVER 0x4: FLC1_ILLMODECH 0x5: FLC1_ILLCMD 0x6: FLC1_ILLSIZE 0x7: FLC1_ILLBANKERASE Note: SSU_ERROR_TYPE.FLC1_ERROR_TYPE register will not get updated on further FLC1 errors until the SSU_ERROR_TYPE.FLC1_ERROR_TYPE bits are cleared. Reset type: PORESETn
8	FLC1_ERROR_STS	R	0h	FLC1_ERROR status Reset type: PORESETn
7	MODE_INVALID	R	0h	BANKMAP, SECVALID, BANKMODE, SSUMODE invalid Reset type: PORESETn
6-4	RESERVED	R	0h	Reserved
3	CPU3_SSU_MMR_ACCESS_ERROR	R	0h	CPU3_SSU_MMR_ACCESS_ERROR Reset type: PORESETn
2	CPU2_SSU_MMR_ACCESS_ERROR	R	0h	CPU2_SSU_MMR_ACCESS_ERROR Reset type: PORESETn
1	CPU1_SSU_MMR_ACCESS_ERROR	R	0h	CPU1_SSU_MMR_ACCESS_ERROR Reset type: PORESETn
0	RESERVED	R	0h	Reserved

8.9.3.113 SSU_ERROR_TYPE_FRC Register (Offset = 88Ch) [Reset = 00000000h]

SSU_ERROR_TYPE_FRC is shown in Figure 8-121 and described in Table 8-138.

Return to the Summary Table.

SSU Error aggregator Error Type Force Register

Figure 8-121. SSU_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	RESERVED	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED						FLC2_MMR_ACCESS_ERROR	FLC2_ERROR_TYPE
R-0h						R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
FLC2_ERROR_TYPE		FLC2_ERROR_STS	FLC1_MMR_ACCESS_ERROR	FLC1_ERROR_TYPE			FLC1_ERROR_STS
R-0/W1S-0h		R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h			R-0/W1S-0h
7	6	5	4	3	2	1	0
MODE_INVALID	RESERVED			CPU3_SSU_MMR_ACCESS_ERROR	CPU2_SSU_MMR_ACCESS_ERROR	CPU1_SSU_MMR_ACCESS_ERROR	RESERVED
R-0/W1S-0h	R-0h			R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h

Table 8-138. SSU_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	RESERVED	R-0/W1S	0h	Reserved
29-18	RESERVED	R	0h	Reserved
17	FLC2_MMR_ACCESS_ERROR	R-0/W1S	0h	FLC2_MMR_ACCESS_ERROR force register 0 - No action 1 - forces corresponding error type flag Reset type: PORESETn
16-14	FLC2_ERROR_TYPE	R-0/W1S	0h	FLC2_ERROR type force register 0 - No action 1 - forces corresponding error type flag Note: SSU_ERROR_TYPE_FRC.FLC2_ERROR_STS should also be written along with writing SSU_ERROR_TYPE_FRC.FLC2_ERROR_TYPE to force any of the FLC2 errors Reset type: PORESETn
13	FLC2_ERROR_STS	R-0/W1S	0h	FLC2_ERROR status force register 0 - No action 1 - forces corresponding error type flag Reset type: PORESETn

Table 8-138. SSU_ERROR_TYPE_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	FLC1_MMR_ACCESS_ERROR	R-0/W1S	0h	FLC1_MMR_ACCESS_ERROR force register 0 - No action 1 - forces corresponding error type flag Reset type: PORESETn
11-9	FLC1_ERROR_TYPE	R-0/W1S	0h	FLC1_ERROR type force register 0 - No action 1 - forces corresponding error type flag Note: SSU_ERROR_TYPE_FRC.FLC1_ERROR_STS should also be written along with writing SSU_ERROR_TYPE_FRC.FLC1_ERROR_TYPE to force any of the FLC1 errors Reset type: PORESETn
8	FLC1_ERROR_STS	R-0/W1S	0h	FLC1_ERROR status force register 0 - No action 1 - forces corresponding error type flag Reset type: PORESETn
7	MODE_INVALID	R-0/W1S	0h	BANKMAP, SECVALID, BANKMODE, SSUMODE invalid force register 0 - No action 1 - forces corresponding error type flag Reset type: PORESETn
6-4	RESERVED	R	0h	Reserved
3	CPU3_SSU_MMR_ACCESS_ERROR	R-0/W1S	0h	CPU3_SSU_MMR_ACCESS_ERROR force register 0 - No action 1 - forces corresponding error type flag Reset type: PORESETn
2	CPU2_SSU_MMR_ACCESS_ERROR	R-0/W1S	0h	CPU2_SSU_MMR_ACCESS_ERROR force register 0 - No action 1 - forces corresponding error type flag Reset type: PORESETn
1	CPU1_SSU_MMR_ACCESS_ERROR	R-0/W1S	0h	CPU1_SSU_MMR_ACCESS_ERROR force register 0 - No action 1 - forces corresponding error type flag Reset type: PORESETn
0	RESERVED	R	0h	Reserved

8.9.3.114 SSU_ERROR_TYPE_CLR Register (Offset = 890h) [Reset = 00000000h]

SSU_ERROR_TYPE_CLR is shown in Figure 8-122 and described in Table 8-139.

Return to the Summary Table.

SSU Error aggregator Error Type Clear Register

Figure 8-122. SSU_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	RESERVED	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED						FLC2_MMR_ACCESS_ERROR	FLC2_ERROR_TYPE
R-0h						R-0/W1C-0h	R-0/W1C-0h
15	14	13	12	11	10	9	8
FLC2_ERROR_TYPE		FLC2_ERROR_STS	FLC1_MMR_ACCESS_ERROR	FLC1_ERROR_TYPE			FLC1_ERROR_STS
R-0/W1C-0h		R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h			R-0/W1C-0h
7	6	5	4	3	2	1	0
MODE_INVALID	RESERVED			CPU3_SSU_MMR_ACCESS_ERROR	CPU2_SSU_MMR_ACCESS_ERROR	CPU1_SSU_MMR_ACCESS_ERROR	RESERVED
R-0/W1C-0h	R-0h			R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0h

Table 8-139. SSU_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	RESERVED	R-0/W1C	0h	Reserved
29-18	RESERVED	R	0h	Reserved
17	FLC2_MMR_ACCESS_ERROR	R-0/W1C	0h	FLC2_MMR_ACCESS_ERROR clear register 0 - No action 1 - Clears corresponding error type flag Reset type: PORESETn
16-14	FLC2_ERROR_TYPE	R-0/W1C	0h	FLC2_ERROR type clear register 0 - No action 1 - Clears corresponding error type flag Note: SSU_ERROR_TYPE_CLR.FLC2_ERROR_STS should also be written along with writing corresponding value in SSU_ERROR_TYPE_CLR.FLC2_ERROR_TYPE to clear the FLC2 errors Reset type: PORESETn
13	FLC2_ERROR_STS	R-0/W1C	0h	FLC2_ERROR status clear register 0 - No action 1 - Clears corresponding error type flag Reset type: PORESETn

Table 8-139. SSU_ERROR_TYPE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	FLC1_MMR_ACCESS_ERROR	R-0/W1C	0h	FLC1_MMR_ACCESS_ERROR clear register 0 - No action 1 - Clears corresponding error type flag Reset type: PORESETn
11-9	FLC1_ERROR_TYPE	R-0/W1C	0h	FLC1_ERROR type clear register 0 - No action 1 - Clears corresponding error type flag Note: SSU_ERROR_TYPE_CLR.FLC1_ERROR_STS should also be written along with writing corresponding value in SSU_ERROR_TYPE_CLR.FLC1_ERROR_TYPE to clear the FLC1 errors Reset type: PORESETn
8	FLC1_ERROR_STS	R-0/W1C	0h	FLC1_ERROR status clear register 0 - No action 1 - Clears corresponding error type flag Reset type: PORESETn
7	MODE_INVALID	R-0/W1C	0h	BANKMAP, SECVALID, BANKMODE, SSUMODE invalid clear register 0 - No action 1 - Clears corresponding error type flag Reset type: PORESETn
6-4	RESERVED	R	0h	Reserved
3	CPU3_SSU_MMR_ACCESS_ERROR	R-0/W1C	0h	CPU3_SSU_MMR_ACCESS_ERROR clear register 0 - No action 1 - Clears corresponding error type flag Reset type: PORESETn
2	CPU2_SSU_MMR_ACCESS_ERROR	R-0/W1C	0h	CPU2_SSU_MMR_ACCESS_ERROR clear register 0 - No action 1 - Clears corresponding error type flag Reset type: PORESETn
1	CPU1_SSU_MMR_ACCESS_ERROR	R-0/W1C	0h	CPU1_SSU_MMR_ACCESS_ERROR clear register 0 - No action 1 - Clears corresponding error type flag Reset type: PORESETn
0	RESERVED	R	0h	Reserved

8.9.3.115 ETHERCAT_HIGHPRIO_ERROR_ADDRESS Register (Offset = 8C0h) [Reset = 00000000h]

ETHERCAT_HIGHPRIO_ERROR_ADDRESS is shown in Figure 8-123 and described in Table 8-140.

Return to the Summary Table.

ETHERCAT Error aggregator High Priority Error address register

Figure 8-123. ETHERCAT_HIGHPRIO_ERROR_ADDRESS Register

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

Table 8-140. ETHERCAT_HIGHPRIO_ERROR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ETHERCAT_HIGHPRIO_ERROR_ADDRESS	R	0h	Address at which first High priority error occurred on ETHERCAT access Reset type: PORESETn

8.9.3.116 ETHERCAT_ERROR_TYPE Register (Offset = 8C8h) [Reset = 00000000h]

ETHERCAT_ERROR_TYPE is shown in Figure 8-124 and described in Table 8-141.

Return to the Summary Table.

ETHERCAT Error aggregator Error Type Register

Figure 8-124. ETHERCAT_ERROR_TYPE Register

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

Table 8-141. ETHERCAT_ERROR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R	0h	high priority error set by software Reset type: PORESETn
30	RESERVED	R	0h	Reserved
29-7	RESERVED	R	0h	Reserved
6	UNCERR	R	0h	Uncorrectable error Reset type: PORESETn
5-0	RESERVED	R	0h	Reserved

8.9.3.117 ETHERCAT_ERROR_TYPE_FRC Register (Offset = 8CCh) [Reset = 00000000h]

ETHERCAT_ERROR_TYPE_FRC is shown in Figure 8-125 and described in Table 8-142.

Return to the Summary Table.

ETHERCAT Error aggregator Error Type Force Register

Figure 8-125. ETHERCAT_ERROR_TYPE_FRC Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	RESERVED	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	UNCERR	RESERVED					
R-0h	R-0/W1S-0h	R-0h					

Table 8-142. ETHERCAT_ERROR_TYPE_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1S	0h	force high priority software error 0 - No action 1 - force high priority software error Reset type: PORESETn
30	RESERVED	R-0/W1S	0h	Reserved
29-7	RESERVED	R	0h	Reserved
6	UNCERR	R-0/W1S	0h	Uncorrectable error flag force register 0 - No action 1 - force Uncorrectable error flag Reset type: PORESETn
5-0	RESERVED	R	0h	Reserved

8.9.3.118 ETHERCAT_ERROR_TYPE_CLR Register (Offset = 8D0h) [Reset = 00000000h]

ETHERCAT_ERROR_TYPE_CLR is shown in Figure 8-126 and described in Table 8-143.

Return to the Summary Table.

ETHERCAT Error aggregator Error Type Clear Register

Figure 8-126. ETHERCAT_ERROR_TYPE_CLR Register

31	30	29	28	27	26	25	24
HP_SOFTWARE_ERR	RESERVED	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	UNCERR	RESERVED					
R-0h	R-0/W1C-0h	R-0h					

Table 8-143. ETHERCAT_ERROR_TYPE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HP_SOFTWARE_ERR	R-0/W1C	0h	Clear high priority software error 0 - No action 1 - clear high priority software error Reset type: PORESETn
30	RESERVED	R-0/W1C	0h	Reserved
29-7	RESERVED	R	0h	Reserved
6	UNCERR	R-0/W1C	0h	Uncorrectable error flag clear register 0 - No action 1 - Clear Uncorrectable error flag Reset type: PORESETn
5-0	RESERVED	R	0h	Reserved

Chapter 9

Flash Module

This chapter describes the architecture and operation of the C29x Flash subsystem.

9.1 Introduction to Flash Memory

Flash is a type of electrically erasable and programmable non-volatile memory. In C2000™ real-time microcontrollers, Flash memory is primarily used to store application code, which executes directly from the internally embedded Flash memory banks. Additionally, Flash can be used to store static data that needs to be preserved between power cycles or device resets.

This chapter describes the structure of Flash memory modules in the device, the proper sequence to initialize and configure Flash, how to improve Flash performance by enabling prefetch and cache mechanisms, and mechanisms for management of firmware updates.

9.1.1 FLASH Related Collateral

Foundational Materials

- C29x Academy - Flash Memory
- Embedded Flash Memory (Video)

Getting Started Materials

- *Serial Flash Programming of C2000 Microcontrollers Application Report*
- [FAQ] FAQ for Flash ECC usage in C2000 devices - Includes ECC test mode, Linker ECC options:
- [FAQ] FAQ on Flash API usage for C2000 devices
- [FAQ] Flash - How to modify an application from RAM configuration to Flash configuration?
- [FAQ] How can we improve the Flash tool performance?
- [FAQ] TI C2000 Device Programming Tools and Services

9.1.2 Features

The features of Flash memory include:

- Up to 8 Flash program code banks, 4 of which can optionally be allocated to CPU3 (if available) at boot time for instruction fetches (For the number and size of Flash banks, refer to the device data manual);
- Up to two Flash controller modules, with up to four program banks per module;
- One dedicated data Flash bank housed in Flash Controller 1 (FLC1), optimized for data bus read accesses;
- Enhanced Flash Read Interface module (FRI), with interleaving of program bank pairs for high performance during 128 bit-wide instruction word fetches;
- Each Flash Controller can program or erase a Flash bank or bank pair while simultaneously reading from the other pair of Flash banks;
- Tight integration with the Safety and Security Unit (SSU) for security management and access control for Flash program, erase and read operations;
- 128-bit-wide Flash programming, with configurable programming options;
- Multiple sectors, with the ability to erase individual/specific sectors while leaving others programmed;
- Dedicated SECCFG Flash regions for storing user security policy settings and boot mode options that are loaded at device start-up;
- BANKMGMT sectors for management of Flash firmware updates using FOTA or Live Firmware Update mechanisms, with anti-rollback protection capability;
- Code prefetch, block cache and data cache mechanisms for enhanced read performance for program code and data;
- Configurable wait states to achieve the best performance at a given clock frequency;
- Safety Features:
  - ECC error detection in address bits, with reporting to the Error Signaling Module (ESM);
  - Supports ECC bits for single error correction and double error detection (SECDED);
- Integrated Flash program and erase state machines in the Flash Controller modules:
  - Simple Flash API algorithms;
  - Fast erase and program times (refer to the device data manual for details);
- Automatic arbitration of data accesses between multiple initiators (C29x CPUs, HSM, RTDMA and debugger accesses).

9.1.3 Flash Tools

Texas Instruments provides the following tools for Flash:

- Code Composer Studio™ (CCS) IDE - the development environment with integrated Flash plugin. TI recommends performing a debug reset and restart after programming the code into Flash using CCS.
- Flash API Library - a set of software peripheral functions to erase/program Flash
- UniFlash - standalone tool to erase/program/verify the Flash content through JTAG. No CCS is required.
- Users must check and install available updates for CCS On-Chip Flash Plugin and UniFlash tools.

9.1.4 Block Diagram

A top-level block diagram of the C29x Flash subsystem is shown in Figure 9-1.

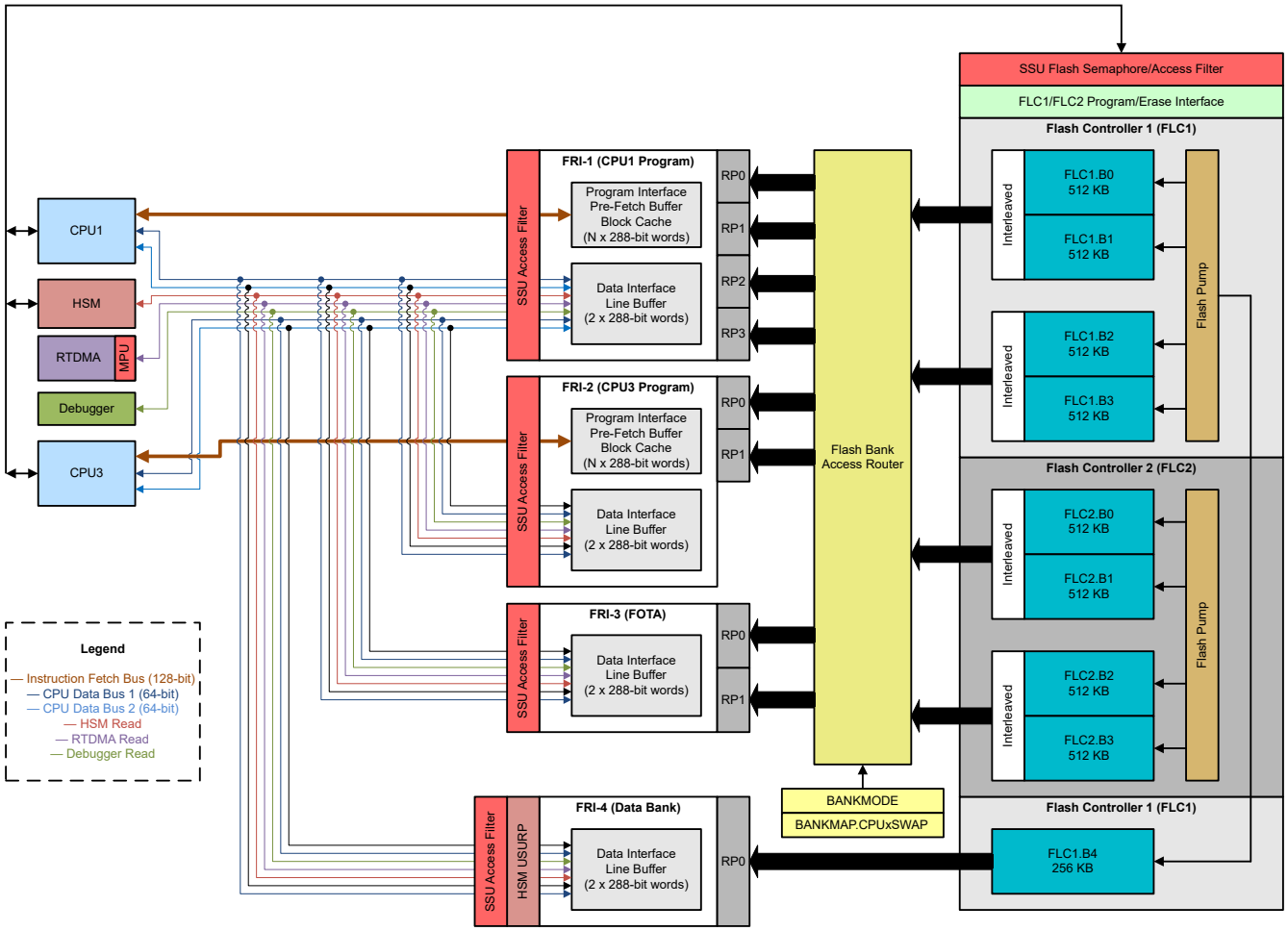


Figure 9-1. Flash Subsystem Block Diagram

9.2 Flash Subsystem Overview

At the heart of the C29x Flash subsystem are one or more Flash Controllers (FLCn), which interface directly with the Flash banks and charge pump. Each Flash controller performs the following primary functions:

- Provides a simple interface for software to program or erase the Flash memory.
- Interfaces with the Flash Bank Access Router (FBAR) for reads of program code and data.
- Provides the capability to block unwanted bank program or erase operations.

Additionally, there are four Flash Read Interfaces (FRI-n), which provide an interface for various initiators in the device to read from Flash memory. The Flash Read Interfaces provide memory address regions for Flash read, and include buffer and cache mechanisms to maximize read performance. The Flash Bank Access Router (FBAR) connects each Flash read port to the associated Flash bank or bank pair, depending on MODE and SWAP configuration. These modules are explained in detail in the following sections. Accesses from all initiators to the Flash Read Interface or Flash Controller registers are filtered through the SSU access permission logic and HSM usurp controls.

9.3 Flash Banks and Pumps

This device includes up to 8 Flash application code banks, and one application data bank. Each Flash bank stores code or data as 128-bit Flash words, plus 16 bits of ECC per Flash word. To meet the performance requirements of the C29 CPU, each group of two code banks is interleaved to form a bank pair, enabling 256-bit instruction fetches. The data bank is a single bank, and does not require interleaving since C29 data buses are 64-bits wide. The F29x Flash API automatically handles the addressing and interleaving of code banks, enabling user code to perform program and erase operations on interleaved bank pairs with a single command.

For naming purposes, each bank or bank pair is referred to by the Flash controller name and bank numbers. Banks are numbered starting with 0. For example, FLC2.B0/B1 refers to the first interleaved bank pair in Flash Controller 2, and FLC1.B4 refers to the 5th non-interleaved bank in FLC1, which is a data bank.

Each bank is divided into the following regions:

1. **MAIN**: This region is the primary storage region for the Flash bank, and is used to store application code and data.
2. **SECCFG**: This region contains security policy settings (also known as the User Protection Policy (UPP)), and other device configuration parameters. Security policy settings are loaded by the boot ROM into the Safety and Security Unit (SSU). For more information and content mapping for the SECCFG region, see Section 10.8.2.
3. **BANKMGMT**: This region is used to manage firmware updates using Firmware-Over-The-Air (FOTA) or Live Firmware Update (LFU) processes. The BANKMGMT sector also contains code version information for rollback protection. For more information on the BANKMGMT region, see Section 10.8.

Each Flash controller also contains a charge pump. The charge pump generates the special voltages and currents required for Flash program, erase and read operations.

9.4 Flash Read Interfaces

The C29x Flash subsystem includes up to four Flash Read Interface modules (FRI-n). The Flash read interface modules provide a means for the various initiators on the system to perform read operations on Flash memory. These initiators include:

- **C29x CPUs**: Each CPU has one 128-bit instruction fetch bus, and two 64-bit data buses. In the C29x architecture, only primary CPUs (odd-numbered) have access to Flash memory for code fetch and data. Secondary CPUs (even-numbered) do not have access to Flash. Code that runs on an independent secondary CPU must first be copied to RAM and then executed from there.
- **RTDMA1**: The primary Real-Time DMA module can perform 64-bit data reads of Flash memory.
- **HSM**: If present, the Hardware Security Module can perform 32-bit data reads of Flash memory through the HSM_TO_SOC bus.
- **Debug Interface**: A debugger connected to either the C29x or HSM CPUs can perform data read accesses to C29x Flash memory locations.

Each Flash read interface is addressable through one or more Flash read ports. Because of bank interleaving, multiple CPU support, and support for bank swapping during firmware updates, Flash code banks do not have a fixed mapping to CPU address space. Rather, each read port has a fixed address range, but can map to a different Flash region depending on the current system configuration. The Flash bank access router performs the background translation, routing read access requests to the intended bank or banks as directed by the Safety and Security Unit (SSU). The are two types of Flash read operations that can be performed: program instruction fetches for CPU execution, and data accesses.

Each of the Flash read interfaces is described in the following sections.

FRI-1: CPU1 Program

FRI-1 interfaces with the instruction fetch bus of the first C29x CPU (CPU1), and is primarily used for executing program code on CPU1. Additionally, FRI-1 is connected to all data read initiators in the system. This read interface has four read ports, numbered RP0 to RP3. These read ports are mapped to code banks depending on the configuration of BANKMODE and BANKMAP.CPU1SWAP registers in the SSU. These settings are described in subsequent sections.

FRI-2: CPU3 Program

On 3- or 4-CPU systems, FRI-2 interfaces with the instruction fetch bus of the third C29x CPU (CPU3). When program Flash memory is allocated to CPU3 by configuring BANKMODE, FRI-2 can be used to execute program code on CPU3, and additionally can be used as a data read interface by all initiators in the system. This read interface has two read ports, numbered RP0 to RP1. These read ports are mapped to code banks depending on the configuration of BANKMODE and BANKMAP.CPU3SWAP. On single-CPU systems, FRI-2 is not present.

FRI-3: Firmware Update Region

FRI-3 provides an address range to be used for programming and verifying an updated firmware image into the second half of Flash memory, while the current application continues to execute from the first half through FRI-1 and FRI-2 (if present). When the firmware update is complete, the Flash memory ranges can then be swapped using BANKMAP.CPUxSWAP, and the newly programmed firmware now executes from FRI-1 and FRI-2 (if present). This supports Firmware-Over-The-Air (FOTA) and Live Firmware Update (LFU) firmware upgrade mechanisms. FRI-3 does not have an instruction fetch interface—only data reads are possible from all system initiators. FRI-3 has two read ports, numbered RP0 to RP1, which are mapped to available Flash code memory.

FRI-4: Data Bank

FRI-4 is exclusively used for the data bank (FLC1.B4), and does not have an instruction fetch interface. Thus, application code cannot be executed from the data bank; only read accesses are possible from all system initiators. FRI-4 has one read port (RP0) which has a fixed mapping to the data Flash bank, and does not change based on the BANKMODE or BANKMAP.CPUxSWAP settings.

9.4.1 Bank Modes and Swapping

The BANKMODE register in the SSU_GEN_REGS register range configures the system Flash bank mode setting. These modes determine the allocation of Flash memory to primary C29 CPUs in the system, and whether the swapping feature is enabled for firmware updates. The BANKMODE register is loaded from the BANKMGMT sector of the active code bank pair in FLC1 during device boot. The active BANKMGMT sector is determined based on the values of the BANK_STATUS and BANK_UPDATE fields; afterward, the winning BANKMGMT sector is used to configure the system BANKMODE.

The available bank mode options are described in Table 9-1.

Table 9-1. C29 Bank Modes

BANKMODE	Flash Mapping	Swap Enabled	1-CPU Devices
0	All program Flash mapped to CPU1	No	Available
1		Yes	Available
2	Program Flash memory is split between CPU1 and CPU3	No	N/A
3		Yes	N/A

The BANKMAP register in SSU_GEN_REGS is used to determine the current active A/B swap setting for firmware updates using FOTA or LFU. This register contains a swap enable bit for each primary CPU (CPUxSWAP).

For specific mappings of Flash address ranges to read port addresses based on BANKMODE and CPUxSWAP settings, refer to the device data sheet.

9.4.2 Flash Wait States

When a CPU performs a read access to a Flash memory address, data is returned after (RWAIT + 1) SYSCLK cycles.

For an access to the SECCFG or BANKMGMT Flash regions, data is returned after 10 SYSCLK cycles.

RWAIT defines the number of random access wait states, and is configured using the RWAIT field in the FRDCNTL register. At reset, RWAIT defaults to a value of 2. RWAIT can be reconfigured to a lower value when the CPU clock frequency is low enough to accommodate the Flash access time. For a table of supported RWAIT values versus CPU clock frequency ranges, refer to the device data sheet.

For a given system clock frequency, configure RWAIT using the following formula:

$$\mathrm{RWAIT} = \mathrm{ceiling}\left(\frac{\mathrm{SYSCLK}}{\mathrm{FCLK}} - 1\right)$$

where SYSCLK is the system operating frequency, and where FCLK is the Flash clock frequency.

FCLK must be ≤ $\mathrm{FCLK}_{\mathrm{max}}$, the allowed maximum Flash clock frequency.

If RWAIT results in a fractional value when calculated using the above formula, round up RWAIT to the nearest integer.

Note

The minimum value that RWAIT can be configured to on this device is 1. Zero wait state reads are not supported.

9.4.3 Buffer and Cache Mechanisms

Each Flash read interface includes mechanisms designed to maximize performance for read operations. These mechanisms include:

- An instruction prefetch buffer,
- A program code block cache, and
- A line buffer for data read operations.

These mechanisms are disabled by default at device boot, and must be configured to enable them before application execution commences.

9.4.3.1 Prefetch Mechanism and Block Cache

Flash memory is typically used to store application code. During code execution, instructions are fetched from contiguous memory addresses, except when a discontinuity occurs. Usually, the portion of code that resides in contiguous address locations makes up the majority of the application code, and is referred to as linear code. To improve the performance of linear code execution, the Flash read interface includes a code prefetch mechanism and block cache. The prefetch mechanism and block cache are available on FRI-1 (and FRI-2, if present). Figure 9-2 shows a functional block diagram of the Flash prefetch mechanism and block cache.

The prefetch mechanism does a look-ahead prefetch on linear address increments, starting from the address of the last instruction fetch. The Flash prefetch mechanism is disabled by default. To enable prefetch mode, set the PREFETCH_EN bit in the FRIx_INTF_CTRL register to 1, or call the Flash_enablePrefetch() driverlib function.

Each instruction fetch from Flash memory reads out 256 bits total—128 bits from each half of an interleaved pair (not counting ECC bits). The starting address of the access from Flash is automatically aligned to a 256-bit boundary, such that the instruction location is within the 256 bits to be fetched. When the prefetch mechanism is enabled, the 256 bits read from the instruction fetch are stored in a 128-bit wide by 4-level deep instruction prefetch buffer. The contents of this prefetch buffer are then sent to the CPU for processing as required.

The C29 CPU receives instruction packets up to 128 bits wide, over a 128-bit program read bus. Each instruction packet can contain a combination of 16-bit, 32-bit or 48-bit instructions. While the instructions are processing through the CPU, the Flash prefetch mechanism automatically initiates another access to the Flash bank to prefetch the next 256 bits. In this manner, the Flash prefetch mechanism works in the background to keep the instruction prefetch buffer as full as possible. Using this technique, the overall efficiency of sequential code execution from Flash is significantly improved.

In addition to the prefetch buffer, the Flash read interface includes a block cache mechanism, consisting of two 256-bit-wide by 16-deep blocks. The block cache sits between the prefetch buffer and the Flash banks, and is loaded with 256-bit data simultaneously with the prefetch buffer. Whenever there is a discontinuity in execution (such as a branch instruction), the mechanism checks if the requested data is already in the block cache. If so, the instruction data is read from the block cache and fed into the prefetch buffer, which then sends the data to the CPU on the next cycle. This improves the latency for discontinuity instructions to just one wait state, boosting code performance in short-branch loops and other minor discontinuities. To enable the block cache, set the CODE_CACHE_EN bit in the FRIx_INTF_CTRL register to 1, or call the Flash_enableCodeCache() driverlib function.

Note

The block cache is designed such that execution loops of 512 bytes or less run with minimum latency once loaded into the cache. Loops that are larger than 512 bytes exceed the size of one cache block, and in these cases, the performance benefit can be reduced.

Because the block cache interfaces between the prefetch buffer and the Flash banks, the prefetch mechanism must be enabled for the block cache to function. If the block cache is enabled without the prefetch buffer, neither mechanism ever gets loaded, and Flash performance is equivalent to both mechanisms being turned off. Always enable the prefetch buffer when enabling the block cache.

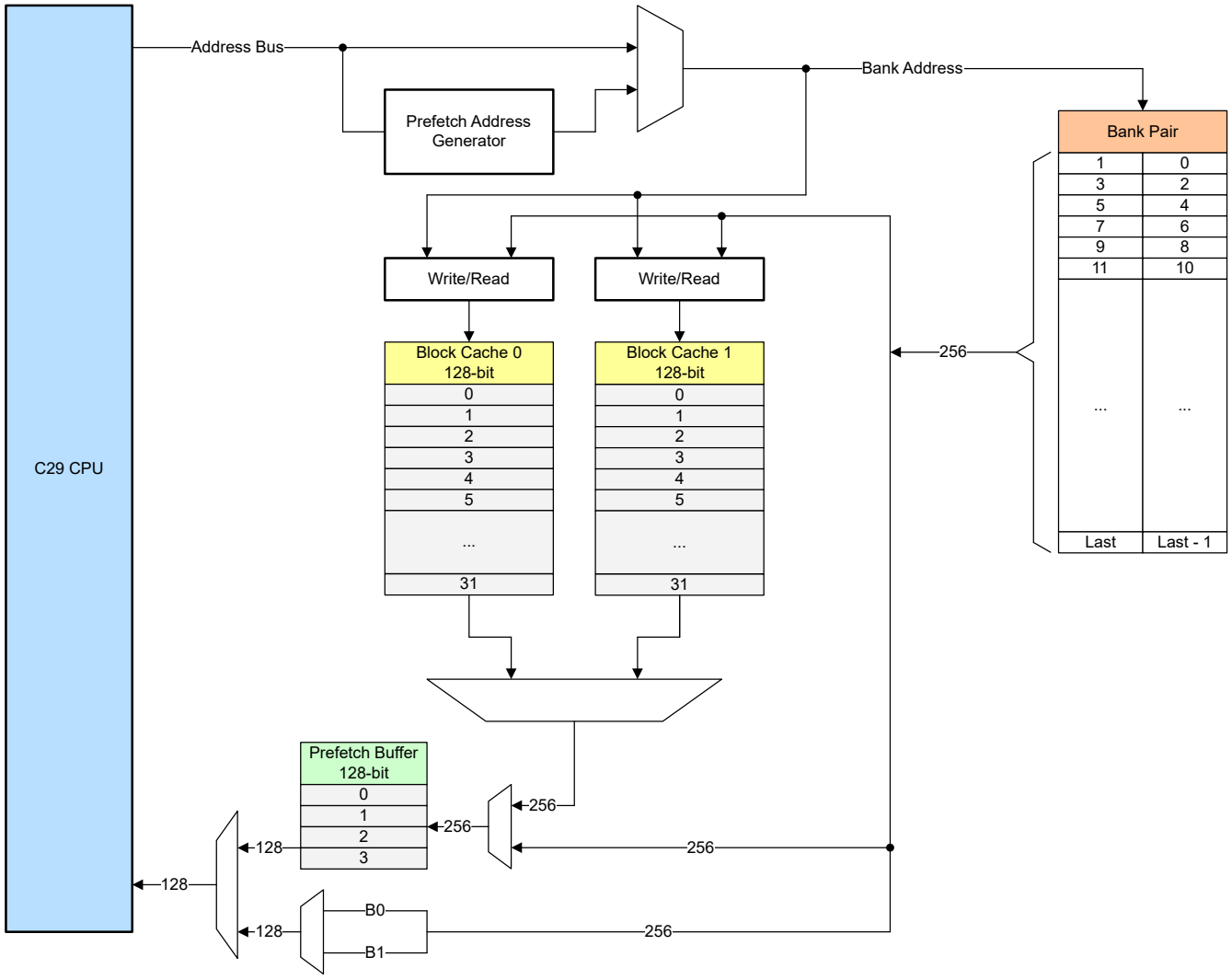


Figure 9-2. Flash Prefetch Mechanism and Block Cache

9.4.3.2 Data Line Buffer

To improve the performance of data reads, the Flash read interface includes a data line buffer. The size of the data line buffer is 288 bits for an interleaved bank pair (including 32 ECC bits), and 144 bits for a non-interleaved data bank (including 16 ECC bits). To activate the Flash data line buffer, write 1 to the DATA_CACHE_EN bit of the FRIx_INTF_CTRL register for the target Flash read interface.

When a read request is received from one or more initiators in the system, the upper portion of the address is compared to the buffer tag to determine if the requested data is in the line buffer. If a match is found, and the valid bit (V) is set, then the Flash read interface returns data directly from the line buffer to the initiator, without making a new access to the Flash bank. The address of the data is also returned to the CPU for ECC checking.

There is a dedicated compare unit for each read initiator (for example, CPU1, RTDMA, debugger), so multiple initiators can read from the data line buffer simultaneously without added wait states. Figure 9-3 shows a functional block diagram of the data line buffer for interleaved bank pairs. Figure 9-4 shows a functional block diagram of the data line buffer for non-interleaved banks.

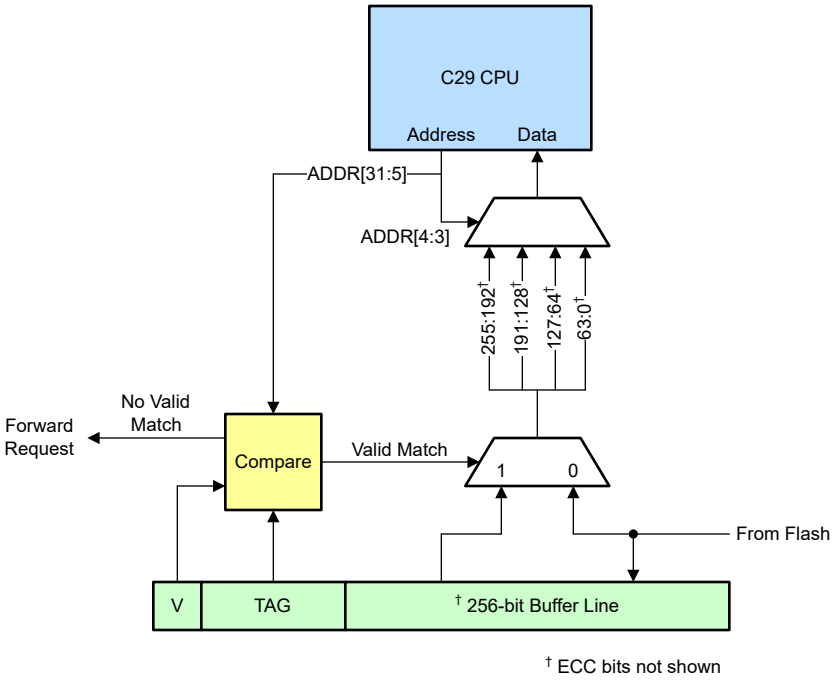


Figure 9-3. Flash Data Line Buffer for Interleaved Banks

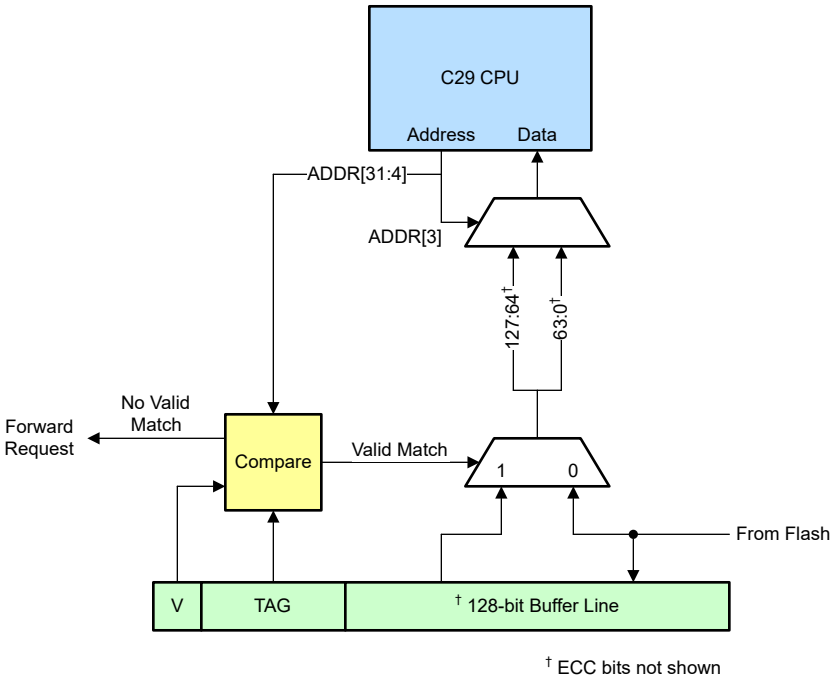


Figure 9-4. Flash Data Line Buffer for Single Bank

If a query does not match the line buffer tag, then a read operation is performed on the associated Flash bank or bank pair. When the data is returned from the Flash, the Flash read interface sends the requested data portion to the CPU, updates the data line buffer, and sets the valid bit. For non-interleaved banks, the line buffer is always 128-bit aligned. For interleaved bank pairs, the line buffer contents are always 256-bit aligned, such that the lower 128 bits come from the first bank and the upper 128 bits come from the second bank. Figure 9-5 illustrates the mapping of interleaved bank pairs to data line buffer bits.

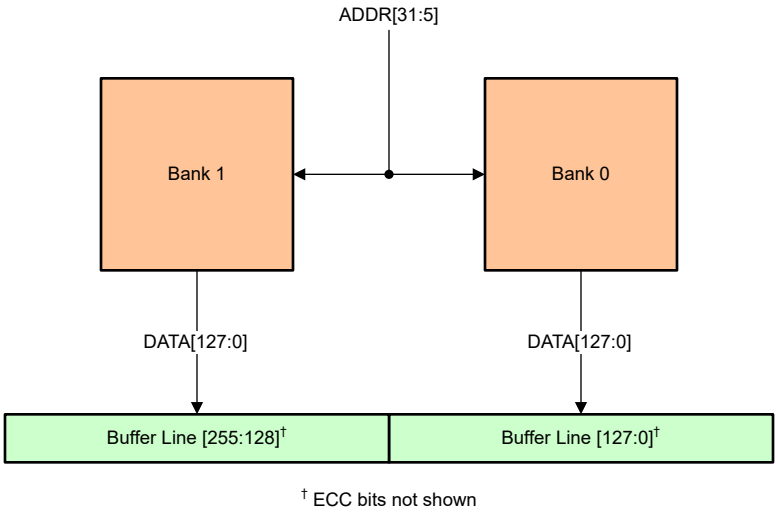


Figure 9-5. Interleaved Bank to Buffer Mapping

Note

Since each Flash Read Interface contains only one buffer, attempts to concurrently read data at different Flash addresses by multiple initiators through the same FRI can potentially lead to a bottleneck. In this situation, the line buffer contents are replaced by each concurrent request without being used. To prevent a slowdown in performance, avoid or limit simultaneous reads by multiple CPUs to the same Flash Read Interface.

The data line buffer valid bit is only cleared automatically on a reset. Always flush the buffer manually after programming or erasing Flash memory, or whenever a CPU fault is detected. To clear the valid bit and flush the buffer, write 1 to the DATA_CACHE_CLR bit of the corresponding FRIx_INTF_CLR register.

Note

The Flash read interface registers can only be written to by CPU1. When programming or erasing Flash memory from another CPU (e.g. CPU3), the application must request CPU1 to flush the data line buffer via IPC.

9.4.3.3 Sequential Data Pre-read Mode

Each Flash read interface provides a user-configurable sequential data pre-read mode. In this mode, the data line automatically pre-reads the Flash and refills the line buffer with the next sequential set of addresses after a previous read is completed. When sequential pre-read mode is active and an initiator reads the last 64-bit word in the line buffer (bits [255:192] for interleaved banks, [127:64] for single banks), the pre-read logic reads the next sequential Flash bank address, and replaces the buffer contents with the data read. The line buffer is then ready for the initiator at the next Flash read request. If an initiator requests a part of the data while the Flash read access is in progress, then the data is returned to the initiator as soon as the Flash read operation is complete.

Flash sequential data pre-read mode is turned off by default, and must be explicitly activated by the user. This mode is intended to be used only during sequential data reads for maximum performance. To enable

sequential data pre-read mode, write 1 to the DATA_PREREAD_EN bit of the FRIx_INTF_CTRL for the target read interface. When the sequential read operation is complete, write 0 to the DATA_PREREAD_EN bit to turn off the pre-read mechanism.

Note

The sequential data pre-read logic is part of the data line buffer, and cannot function if the line buffer is turned off. Always enable the line buffer when using sequential pre-read mode.

9.4.4 Flash Read Arbitration

Each Flash read interface contains multiple levels of arbitration to determine in which order concurrent read requests from multiple initiators are serviced.

1. The first level of arbitration is a fixed priority arbiter between C29 CPU buses. Data Read Bus 1 (DRB1) always gets a higher priority than Data Read Bus 2 (DRB2). This arbitration level enables zero-wait-state switching between the two data buses belonging to the same CPU. The actual data read bus used for a specific read request is automatically generated by the compiler.
2. The second level of arbitration is the L1 data read pipeline. This level arbitrates between data read requests generated by multiple initiators in the system. The initiators are selected using a round-robin pointer. The round-robin selection order for FRI-1, FRI-3, and FRI-4 is as follows:
   a. CPU1
   b. CPU2 (if present)
   c. CPU3 (if present)
   d. CPU4 (if present)
   e. RTDMA
   f. HSM (if present)
   g. Data Pre-read
   h. Debugger

   The round-robin selection order for FRI-2, if present, is as follows:
   a. CPU3
   b. CPU4 (if present)
   c. CPU1
   d. CPU2 (if present)
   e. RTDMA
   f. HSM (if present)
   g. Data Pre-read
   h. Debugger

   Switching between initiators incurs a one-cycle pipeline delay, to allow for the round-robin pointer change.
3. For FRI-1 (and FRI-2, when present), there is a third level of arbitration between fetch and data requests (L2). This arbiter is also a round-robin arbiter like the second level, selecting using the following order:
   a. Instruction fetch (CPU1 for FRI-1, CPU3 for FRI-2)
   b. L1 data read pipeline output

   Changes to the L2 round-robin pointer incur a one-cycle pipeline delay, similar to the L1 arbiter.

9.4.5 Error Correction Code (ECC) Protection

F29x platform devices feature an end-to-end safety architecture, with ECC logic built into the C29 CPU. When a Flash data read or instruction fetch operation is performed, ECC check bits are sent along with the data bits onto the CPU program or data buses. The CPU automatically checks incoming instructions and data using the ECC bits, correcting single-bit data errors, and generating a fault for double-bit (uncorrectable) errors. These errors generate events to the Error Signaling Module (ESM), which in turn can generate a regular interrupt for a single-bit error, or a non-masking interrupt (NMI) for a double-bit error.

Each 8-bit ECC code is computed based on the requested data address and 64 bits of data. For each 128-bit Flash word, there are two ECC codes (upper 64 bits and lower 64 bits), for a total of 16 ECC bits. ECC bits can be independently read through separate FRI read port memory regions as described in the device data sheet.

For more information on ECC protections, see Chapter 3.

Note

Because the ECC logic is built into the C29 CPU, read accesses from other initiators such as the debugger, RTDMA or HSM CPU do not have ECC protection. Flash reads from these initiators do not have error detection and correction enabled.

9.4.6 Procedure to Change Flash Read Interface Registers

While configuring a Flash Read Interface (FRI), no accesses to any Flash memory that is covered by the Flash Read Interface can be in progress. This includes instructions still in the CPU pipeline, data reads, and instruction prefetch operations. To be sure that no access takes place during the configuration change, follow the procedure shown below for any code that modifies the Flash Read Interface registers.

Note

Flash Read Interface registers can only be modified by code running as CPU1.LINK2, CPU1.LINK1, or the HSM CPU. All other LINKs have read-only access to these registers.

1. Start executing the application code from RAM or Flash.
2. Branch to or call the Flash configuration code (that writes the FRI registers) in RAM. This is required to properly flush the CPU pipeline before the configuration change. Any function that changes the FRI configuration must reside in RAM, and cannot execute from Flash memory using the same FRI.
3. Execute the Flash configuration code to configure the FRI registers (FRDCNTL, FLCLKCTL, FRIx_INTF_CTRL, and so on).
4. At the end of the Flash configuration code execution, wait nine cycles to allow the write instructions to propagate through the CPU pipeline. This must be done before the return-from-function call is made.
5. Return to the calling function residing in Flash or RAM, and continue execution.

Note

Flash read interfaces that do not include instruction fetch capability, such as FRI-3 and FRI-4, can be configured by code executing directly from Flash, provided that no data access to the covered Flash banks is being made. In such cases, verify that no current data access to the target Flash banks is being performed by placing the FRI configuration code in a separate function. Be sure to wait nine cycles to flush the CPU pipeline before branching back to regular application execution.

9.5 Flash Erase and Program

Flash memory can be programmed either by using the CCS Flash plug-in or by using the UniFlash application. If these methods are not feasible in an application, the Flash API can be used. The Flash memory can be programmed, erased, and verified only by using the Flash API library. These functions are written, compiled and validated by Texas Instruments. The Flash module contains a Flash state machine (FSM) to perform program and erase operations.

The recommended flow for programming Flash is:
Erase → Program → Verify

9.5.1 Flash Semaphore and Update Protection

Before program and erase operations can be performed, the currently running application must grab the Flash semaphore. The Flash semaphore is managed by the Safety and Security Unit, and can be grabbed by writing to the FLSEMREQ SSU register. For more information on how to use the Flash semaphore, see Section 10.9.

The device security architecture also provides mechanisms to protect Flash banks and sectors from program and erase operations for safety and security. For more on these protection mechanisms, see Section 10.9.1.

9.5.2 Erase

An erase operation causes all bits in the target Flash memory region to read as 1. A previously programmed Flash word (128 bits) must first be erased before new data can be programmed into the Flash word. The minimum memory size that can be erased in one operation is a Flash sector; single bits or words cannot be erased individually.

Erase operations are performed by the Flash State Machine (FSM) inside the Flash controller. All bits in each target sector are erased together, including data and ECC bits. Erase operations must be initiated using the F29x Flash API, or a GUI front-end such as the Code Composer Studio Flash plug-in or UniFlash. The Flash API always erases sectors from both banks of an interleaved pair when an erase command is issued. When using the Flash API, erase is a non-blocking operation, so the application can perform other tasks while waiting for an erase command to complete.

Bank erase operations are also supported on this device. When performing bank erase, special sectors such as SECCFG and BANKMGMT are not included in the erase operation and must be erased separately.

Note

Because of the physical nature of Flash erase operations, the amount of time taken to complete an erase command gradually increases over time as more write-erase cycles are performed. There is also a lifetime limit on the number of write-erase cycles that are possible for each Flash Controller on a single device. For details on erase timing and write-erase cycle limits, see the device data sheet.

9.5.3 Program

A program operation modifies individual bits in a Flash word to read as 0 instead of 1. A program operation can only be performed on a previously erased bit. Flash programming must be aligned to 64-bit address boundaries, and each 64-bit word can only be programmed once per write/erase cycle.

The Flash State Machine provides a command to program the available user Flash regions: main Flash, SECCFG, and BANKMGMT. This command is also used to program the Error Correction Code (ECC) check bits. Program operations must be initiated using the F29x Flash API, or a GUI front-end such as the Code Composer Studio Flash plug-in or UniFlash. The API provides the ability to automatically generate and program ECC check bits together with the data being programmed. Programming is a non-blocking operation, so the application can perform other tasks while waiting for programming to complete.

After programming, the user must perform a verify operation on the programmed Flash region using the API function Fapi_doVerify(), or using one of the Flash API GUI front-ends. This function verifies the Flash contents against the supplied data.

On this device, any of the primary application CPUs (CPU1, CPU3) or the HSM CPU can program any of the Flash banks on the device after grabbing the Flash semaphore. Secondary CPUs such as CPU2 do not have access to Flash memory. For security, programming access can be restricted to authorized code only by configuring the Flash update owner LINK and CPU in the SSU. Additionally, specific sectors can be completely blocked from program and erase using write/erase protection bits (WEPROT). For more information on these protection mechanisms, see Section 10.9.1.

9.6 Migrating an Application from RAM to Flash

To migrate an existing application that is configured to run from RAM to a Flash-based linker configuration, follow these steps:

1. Replace the RAM linker command file with a Flash linker command file. For examples of Flash-based linker command files, see the `device_support\<device>\common\cmd` directory.
2. When modifying the Flash-based linker command file, be sure to map any initialized sections to Flash memory regions.
3. Make sure the boot mode pins are configured for Flash boot. This tells the boot ROM to redirect execution to the application programmed into Flash memory after boot code execution is complete. For more information on boot mode configuration, see *Detailed Description* > *Device Boot Modes* in the device data sheet.
4. When the device is configured for Flash boot, the boot ROM redirects execution to the Flash entry point location (defined as `BEGIN` in TI-provided Flash linker command files) at the end of boot code execution. Make sure there is a branch instruction at the Flash entry point to your code initialization (for example, `_c_int00`) function. In the C2000Ware examples, the entry point code is specified in the `codestartbranch.asm` file.
5. To achieve best performance for Flash execution, configure the Flash wait states as per the device operating clock frequency, as specified in the device data sheet. In addition, enable prefetch mode and data cache mode. Calling the `Flash_initModule()` driverlib function achieves these steps. Note that code that initializes the Flash module must execute from a RAM location. This is accomplished by assigning the Flash initialization function to the `.TI.ramfunc` section. In the linker command file, map this section to Flash for load, and RAM for execution. The example cmd files provided in C2000Ware show how to do this correctly.
6. For any functions that require 0- or 1-wait state performance, be sure to map to RAM for execution in the linker command file, similar to the Flash initialization function. The `.TI.ramfunc` section in the TI-provided Flash linker command files accomplishes this purpose.
7. Align all code and data sections to 128-bit address boundaries when mapping to Flash memory, using the `ALIGN` directive in the linker command file.
8. For EABI executable formats, define all uninitialized sections mapped to RAM as NOINIT sections (using the directive "`type=NOINIT`") in the linker command file.
9. Be sure to program ECC bits correctly for the Flash application image. Keep the AutoEccGeneration option enabled in the Code Composer Studio Flash Plugin or UniFlash GUI.

9.7 Flash Registers

This section describes the Flash Module Registers.

9.7.1 FLASH Base Address Table

Table 9-2. FLASH Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
FLASH_CMD_REGS_FLC1	FLASHCONTROLLER1_BASE	0x3010_0000	-	YES	-	YES	-	-	YES	YES
FLASH_CMD_REGS_FLC2	FLASHCONTROLLER2_BASE	0x3011_0000	-	YES	-	YES	-	-	YES	YES
FRI_CTRL_REGS [1]	FRI1_BASE	0x301D_0000	-	YES	YES	YES	-	-	YES	YES

(1) Registers writeable by CPU1.LINK0, CPU1.LINK1, CPU1.LINK2 only. All CPUs can read all registers in all LINKs. Debug write access only allowed if Zone0 or Zone1 are enabled for full debug by all CPUs. Debug reads always allowed. Register Read/Write access by HSM.

9.7.2 FLASH_CMD_REGS_FLC1 Registers

Table 9-3 lists the memory-mapped registers for the FLASH_CMD_REGS_FLC1 registers. All register offset addresses not listed in Table 9-3 should be considered as reserved locations and the register contents should not be modified.

Table 9-3. FLASH_CMD_REGS_FLC1 Registers

Offset	Acronym	Register Name
1D0h	CMDWEPROTA	Command Write Erase Protect A Register
1D4h	CMDWEPROTB	Command Write Erase Protect B Register
210h	CMDWEPROTNM	Command Write Erase Protect Non-Main Register
3D0h	STATCMD	Command Status Register

Complex bit access types are encoded to fit into small table cells. Table 9-4 shows the codes that are used for access types in this section.

Table 9-4. FLASH_CMD_REGS_FLC1 Access Type Codes

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

9.7.2.1 CMDWEPROTA Register (Offset = 1D0h) [Reset = 00000000h]

CMDWEPROTA is shown in Figure 9-6 and described in Table 9-5.

Return to the Summary Table.

Command WriteErase Protect A Register
This register allows the first 32 sectors of the main region to be protected from program or erase, with 1 bit protecting each sector. If the main region size is smaller than 32 sectors, then this register provides protection for the whole region.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
In addition, this register is used to aggregate masking for sectors that do not require additional erase pulses during bank erase operations, and will be written to all 1 after the completion of all flash wrapper commands.

Figure 9-6. CMDWEPROTA Register

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

Table 9-5. CMDWEPROTA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	Each bit protects 1 sector. bit [0]: When 1, sector 0 of the flash memory will be protected from program and erase. bit [1]: When 1, sector 1 of the flash memory will be protected from program and erase. : : bit [31]: When 1, sector 31 of the flash memory will be protected from program and erase. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

9.7.2.2 CMDWEPROTB Register (Offset = 1D4h) [Reset = 00000000h]

CMDWEPROTB is shown in Figure 9-7 and described in Table 9-6.

Return to the Summary Table.

Command WriteErase Protect B Register
This register allows main region sectors to be protected from program and erase. Each bit corresponds to a group of 8 sectors.
There are 3 cases for how these protect bits are applied:
1. Single-bank system:
In the case where only a single flash bank is present, the first 32 sectors are protected via the CMDWEPROTA register. Thus, the protection give by the bits in CMDWEPROTB begin with sector 32.
2. Multi-bank system, Bank 0:
When multiple flash banks are present, the first 32 sectors of bank 0 are protected via the CMDWEPROTA register. Thus, only bits 4 and above of CMDWEPROTB would be applicable to bank 0. The protection of bit 4 and above would begin at sector 32. Bits 3:0 of WEPROTB are ignored for bank 0.
3. Multi-bank system, Banks 1-N:
For banks other than bank 0 in a multi-bank system, CMDWEPROTA has no effect, so the bits in CMDWEPROTB will protect these banks starting from sector 0.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
In addition, this register is used to aggregate masking for sectors that do not require additional erase pulses during bank erase operations, and will be written to all 1 after the completion of all flash wrapper commands.

Figure 9-7. CMDWEPROTB Register

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

Table 9-6. CMDWEPROTB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	Each bit protects a group of 8 sectors. When a bit is 1, the associated 8 sectors in the flash will be protected from program and erase. A maximum of 256 sectors can be protected with this register. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

9.7.2.3 CMDWEPROTNM Register (Offset = 210h) [Reset = 00000000h]

CMDWEPROTNM is shown in Figure 9-8 and described in Table 9-7.

Return to the Summary Table.

Command WriteErase Protect Non-Main
Register
This register allows non-main region region sectors to be protected from program and erase. Each bit corresponds to 1 sector.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
In addition, this register is used to aggregate masking for sectors that do not require additional erase pulses during bank erase operations, and will be written to all 1 after the completion of all flash wrapper commands.

Figure 9-8. CMDWEPROTNM Register

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

Table 9-7. CMDWEPROTNM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	Each bit protects 1 sector. bit [0]: When 1, sector 0 of the non-main region will be protected from program and erase. bit [1]: When 1, sector 1 of the non-main region will be protected from program and erase. : : bit [31]: When 1, sector 31 of the non-main will be protected from program and erase. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

9.7.2.4 STATCMD Register (Offset = 3D0h) [Reset = 00000000h]

STATCMD is shown in Figure 9-9 and described in Table 9-8.

Return to the Summary Table.

Command Status Register This register contains status regarding completion and errors of command execution.

Figure 9-9. STATCMD Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED			FAILMISC	RESERVED			FAILINVDATA
R-0h			R-0h	R-0h			R-0h
7	6	5	4	3	2	1	0
FAILMODE	FAILILLADDR	FAILVERIFY	FAILWEPROT	RESERVED	CMDINPROGRESS	CMDPASS	CMDDONE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-8. STATCMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	Reserved
12	FAILMISC	R	0h	Command failed due to error other than write/erase protect violation or verify error. This is an extra bit in case a new failure mechanism is added which requires a status bit. 0h = No Fail 1h = Fail
11-9	RESERVED	R	0h	Reserved
8	FAILINVDATA	R	0h	Program command failed because an attempt was made to program a stored 0 value to a 1. 0h = No Fail 1h = Fail
7	FAILMODE	R	0h	Command failed because a bank has been set to a mode other than READ. Program and Erase commands cannot be initiated unless all banks are in READ mode. 0h = No Fail 1h = Fail
6	FAILILLADDR	R	0h	Command failed due to the use of an illegal address 0h = No Fail 1h = Fail
5	FAILVERIFY	R	0h	Command failed due to verify error 0h = No Fail 1h = Fail
4	FAILWEPROT	R	0h	Command failed due to Write/Erase Protect Sector Violation 0h = No Fail 1h = Fail
3	RESERVED	R	0h	Reserved
2	CMDINPROGRESS	R	0h	Command In Progress 0h = Complete 1h = In Progress

Table 9-8. STATCMD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	CMDPASS	R	0h	Command Pass - valid when CMD_DONE field is 1 0h = Fail 1h = Pass
0	CMDDONE	R	0h	Command Done 0h = Not Done 1h = Done

9.7.3 FLASH_CMD_REGS_FLC2 Registers

Table 9-9 lists the memory-mapped registers for the FLASH_CMD_REGS_FLC2 registers. All register offset addresses not listed in Table 9-9 should be considered as reserved locations and the register contents should not be modified.

Table 9-9. FLASH_CMD_REGS_FLC2 Registers

Offset	Acronym	Register Name
1D0h	CMDWEPROTA	Command Write Erase Protect A Register
1D4h	CMDWEPROTB	Command Write Erase Protect B Register
210h	CMDWEPROTNM	Command Write Erase Protect Non-Main Register
3D0h	STATCMD	Command Status Register

Complex bit access types are encoded to fit into small table cells. Table 9-10 shows the codes that are used for access types in this section.

Table 9-10. FLASH_CMD_REGS_FLC2 Access Type Codes

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

9.7.3.1 CMDWEPROTA Register (Offset = 1D0h) [Reset = 00000000h]

CMDWEPROTA is shown in Figure 9-10 and described in Table 9-11.

Return to the Summary Table.

Command WriteErase Protect A Register
This register allows the first 32 sectors of the main region to be protected from program or erase, with 1 bit protecting each sector. If the main region size is smaller than 32 sectors, then this register provides protection for the whole region.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
In addition, this register is used to aggregate masking for sectors that do not require additional erase pulses during bank erase operations, and will be written to all 1 after the completion of all flash wrapper commands.

Figure 9-10. CMDWEPROTA Register

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

Table 9-11. CMDWEPROTA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	Each bit protects 1 sector. bit [0]: When 1, sector 0 of the flash memory will be protected from program and erase. bit [1]: When 1, sector 1 of the flash memory will be protected from program and erase. : : bit [31]: When 1, sector 31 of the flash memory will be protected from program and erase. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

9.7.3.2 CMDWEPROTB Register (Offset = 1D4h) [Reset = 00000000h]

CMDWEPROTB is shown in Figure 9-11 and described in Table 9-12.

Return to the Summary Table.

Command WriteErase Protect B Register
This register allows main region sectors to be protected from program and erase. Each bit corresponds to a group of 8 sectors.
There are 3 cases for how these protect bits are applied:
1. Single-bank system:
In the case where only a single flash bank is present, the first 32 sectors are protected via the CMDWEPROTA register. Thus, the protection give by the bits in CMDWEPROTB begin with sector 32.
2. Multi-bank system, Bank 0:
When multiple flash banks are present, the first 32 sectors of bank 0 are protected via the CMDWEPROTA register. Thus, only bits 4 and above of CMDWEPROTB would be applicable to bank 0. The protection of bit 4 and above would begin at sector 32. Bits 3:0 of WEPROTB are ignored for bank 0.
3. Multi-bank system, Banks 1-N:
For banks other than bank 0 in a multi-bank system, CMDWEPROTA has no effect, so the bits in CMDWEPROTB will protect these banks starting from sector 0.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
In addition, this register is used to aggregate masking for sectors that do not require additional erase pulses during bank erase operations, and will be written to all 1 after the completion of all flash wrapper commands.

Figure 9-11. CMDWEPROTB Register

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

Table 9-12. CMDWEPROTB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	Each bit protects a group of 8 sectors. When a bit is 1, the associated 8 sectors in the flash will be protected from program and erase. A maximum of 256 sectors can be protected with this register. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

9.7.3.3 CMDWEPROTNM Register (Offset = 210h) [Reset = 00000000h]

CMDWEPROTNM is shown in Figure 9-12 and described in Table 9-13.

Return to the Summary Table.

Command WriteErase Protect Non-Main
Register
This register allows non-main region region sectors to be protected from program and erase. Each bit corresponds to 1 sector.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
In addition, this register is used to aggregate masking for sectors that do not require additional erase pulses during bank erase operations, and will be written to all 1 after the completion of all flash wrapper commands.

Figure 9-12. CMDWEPROTNM Register

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

Table 9-13. CMDWEPROTNM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	Each bit protects 1 sector. bit [0]: When 1, sector 0 of the non-main region will be protected from program and erase. bit [1]: When 1, sector 1 of the non-main region will be protected from program and erase. : : bit [31]: When 1, sector 31 of the non-main will be protected from program and erase. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

9.7.3.4 STATCMD Register (Offset = 3D0h) [Reset = 00000000h]

STATCMD is shown in Figure 9-13 and described in Table 9-14.

Return to the Summary Table.

Command Status Register This register contains status regarding completion and errors of command execution.

Figure 9-13. STATCMD Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED			FAILMISC	RESERVED			FAILINVDATA
R-0h			R-0h	R-0h			R-0h
7	6	5	4	3	2	1	0
FAILMODE	FAILILLADDR	FAILVERIFY	FAILWEPROT	RESERVED	CMDINPROGRESS	CMDPASS	CMDDONE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-14. STATCMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	Reserved
12	FAILMISC	R	0h	Command failed due to error other than write/erase protect violation or verify error. This is an extra bit in case a new failure mechanism is added which requires a status bit. 0h = No Fail 1h = Fail
11-9	RESERVED	R	0h	Reserved
8	FAILINVDATA	R	0h	Program command failed because an attempt was made to program a stored 0 value to a 1. 0h = No Fail 1h = Fail
7	FAILMODE	R	0h	Command failed because a bank has been set to a mode other than READ. Program and Erase commands cannot be initiated unless all banks are in READ mode. 0h = No Fail 1h = Fail
6	FAILILLADDR	R	0h	Command failed due to the use of an illegal address 0h = No Fail 1h = Fail
5	FAILVERIFY	R	0h	Command failed due to verify error 0h = No Fail 1h = Fail
4	FAILWEPROT	R	0h	Command failed due to Write/Erase Protect Sector Violation 0h = No Fail 1h = Fail
3	RESERVED	R	0h	Reserved
2	CMDINPROGRESS	R	0h	Command In Progress 0h = Complete 1h = In Progress

Table 9-14. STATCMD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	CMDPASS	R	0h	Command Pass - valid when CMD_DONE field is 1 0h = Fail 1h = Pass
0	CMDDONE	R	0h	Command Done 0h = Not Done 1h = Done

9.7.4 FRI_CTRL_REGS Registers

Table 9-15 lists the memory-mapped registers for the FRI_CTRL_REGS registers. All register offset addresses not listed in Table 9-15 should be considered as reserved locations and the register contents should not be modified.

Table 9-15. FRI_CTRL_REGS Registers

Offset	Acronym	Register Name	Protection
0h	REVISION	IP Revision Register	
10h	FRDCNTL	Flash Read Control Register	PARITY_PROTECTED
14h	FRDCNTL_LOCK	Flash Read Control Lock Register	PARITY_PROTECTED
18h	FRDCNTL_COMMIT	Flash Read Control Commit Register	PARITY_PROTECTED
30h	FRI1_INTF_CTRL	Flash Read Interface 1 Control Register	PARITY_PROTECTED
34h	FRI1_INTF_CTRL_LOCK	Flash Read Interface 1 Control Lock Register	PARITY_PROTECTED
38h	FRI1_INTF_CTRL_COMMIT	Flash Read Interface 1 Control Commit Register	PARITY_PROTECTED
3Ch	FRI1_INTF_CLR	Flash Read Interface 1 Clear Register	PARITY_PROTECTED
40h	FRI2_INTF_CTRL	Flash Read Interface 2 Control Register	PARITY_PROTECTED
44h	FRI2_INTF_CTRL_LOCK	Flash Read Interface 2 Control Lock Register	PARITY_PROTECTED
48h	FRI2_INTF_CTRL_COMMIT	Flash Read Interface 2 Control Commit Register	PARITY_PROTECTED
4Ch	FRI2_INTF_CLR	Flash Read Interface 2 Clear Register	PARITY_PROTECTED
50h	FRI3_INTF_CTRL	Flash Read Interface 3 Control Register	PARITY_PROTECTED
54h	FRI3_INTF_CTRL_LOCK	Flash Read Interface 3 Control Lock Register	PARITY_PROTECTED
58h	FRI3_INTF_CTRL_COMMIT	Flash Read Interface 3 Control Commit Register	PARITY_PROTECTED
5Ch	FRI3_INTF_CLR	Flash Read Interface 3 Clear Register	PARITY_PROTECTED
60h	FRI4_INTF_CTRL	Flash Read Interface 4 Control Register	PARITY_PROTECTED
64h	FRI4_INTF_CTRL_LOCK	Flash Read Interface 4 Control Lock Register	PARITY_PROTECTED
68h	FRI4_INTF_CTRL_COMMIT	Flash Read Interface 4 Control Commit Register	PARITY_PROTECTED
6Ch	FRI4_INTF_CLR	Flash Read Interface 4 Clear Register	PARITY_PROTECTED
70h	PARITY_TEST	Parity Test Enable	
74h	PARITY_TEST_LOCK	Parity Test Lock Register	
78h	PARITY_TEST_COMMIT	Parity Test Commit Register	

Complex bit access types are encoded to fit into small table cells. Table 9-16 shows the codes that are used for access types in this section.

Table 9-16. FRI_CTRL_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

9.7.4.1 REVISION Register (Offset = 0h) [Reset = 00000000h]

REVISION is shown in Figure 9-14 and described in Table 9-17.

Return to the Summary Table.

IP Revision Register

Figure 9-14. REVISION Register

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

Table 9-17. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	RESERVED	R	0h	Reserved
15-8	MAJREV	R	0h	This hardcoded field defines the major revision of the IP. Reset type: XRSn
7-0	MINREV	R	0h	This hardcoded field defines the minor revision of the IP. Reset type: XRSn

9.7.4.2 FRDCNTL Register (Offset = 10h) [Reset = 02000200h]

FRDCNTL is shown in Figure 9-15 and described in Table 9-18.

Return to the Summary Table.

Flash Read Control Register

Figure 9-15. FRDCNTL Register

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

Table 9-18. FRDCNTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Reserved
27-24	TRIMENGRRWAIT	R/W	2h	Read waitstates of Trim and ENGR sectors. These bits indicate how many waitstates are added to a flash read/fetch access. The TRIMENGRRWAIT value can be set anywhere from 1 to 0xF (zero waitstates is not supported). For a flash access from TRIM or ENGR sectors, data is returned in TRIMENGRRWAIT+1 SYSCLK cycles. Note: The required wait states for each SYSCLK frequency can be found in the device data manual. Reset type: XRSn
23-16	RESERVED	R	0h	Reserved
15-12	RESERVED	R	0h	Reserved
11-8	RWAIT	R/W	2h	Read waitstates. These bits indicate how many waitstates are added to a flash read/fetch access. The RWAIT value can be set anywhere from 1 to 0xF (zero waitstates is not supported). For a flash access to any non-TRIM or ENGR section, data is returned in RWAIT+1 SYSCLK cycles. Note: The required wait states for each SYSCLK frequency can be found in the device data manual. Reset type: XRSn
7-0	RESERVED	R	0h	Reserved

9.7.4.3 FRDCNTL_LOCK Register (Offset = 14h) [Reset = 00000000h]

FRDCNTL_LOCK is shown in Figure 9-16 and described in Table 9-19.

Return to the Summary Table.

Flash Read Control Lock Register

Figure 9-16. FRDCNTL_LOCK Register

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

Table 9-19. FRDCNTL_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-1	RESERVED	R-0	0h	Reserved
0	LOCK	R/W	0h	Determines whether the FRDCNTL register can be written. 0 : Register can be written 1 : Register cannot be written This bit can only be modified if FRDCNTL_COMMIT.COMMIT is zero. Reset type: XRSn

9.7.4.4 FRDCNTL_COMMIT Register (Offset = 18h) [Reset = 00000000h]

FRDCNTL_COMMIT is shown in Figure 9-17 and described in Table 9-20.

Return to the Summary Table.

Flash Read Control Commit Register

Figure 9-17. FRDCNTL_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 9-20. FRDCNTL_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the FRDCNTL_LOCK register. This bit cannot be cleared, except by reset. 0 : FRDCNTL_LOCK is modifiable 1 : FRDCNTL_LOCK is committed permanently Reset type: XRSn

9.7.4.5 FRI1_INTF_CTRL Register (Offset = 30h) [Reset = 00000000h]

FRI1_INTF_CTRL is shown in Figure 9-18 and described in Table 9-21.

Return to the Summary Table.

Flash Read Interface 1 Control Register

Figure 9-18. FRI1_INTF_CTRL Register

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

Table 9-21. FRI1_INTF_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	RESERVED	R	0h	Reserved
3	DATA_PREREAD_EN	R/W	0h	Sequential data preread mode enable. 0 : A value of 0 disables the mode. 1 : A value of 1 enables the mode. Reset type: XRSn
2	CODE_CACHE_EN	R/W	0h	Code block cache enable. 0 : A value of 0 disables the code cache. 1 : A value of 1 enables the code cache. Reset type: XRSn
1	DATA_CACHE_EN	R/W	0h	Data line buffer enable. 0 : A value of 0 disables the data line buffer. 1 : A value of 1 enables the data line buffer. Reset type: XRSn
0	PREFETCH_EN	R/W	0h	Prefetch enable. 0 : A value of 0 disables prefetch mechanism. 1 : A value of 1 enables prefetch mechanism. Reset type: XRSn

9.7.4.6 FRI1_INTF_CTRL_LOCK Register (Offset = 34h) [Reset = 00000000h]

FRI1_INTF_CTRL_LOCK is shown in Figure 9-19 and described in Table 9-22.

Return to the Summary Table.

Flash Read Interface 1 Control Lock Register

Figure 9-19. FRI1_INTF_CTRL_LOCK Register

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

Table 9-22. FRI1_INTF_CTRL_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-1	RESERVED	R-0	0h	Reserved
0	LOCK	R/W	0h	Determines whether the FRD_INTF_CTRL register can be written. 0 : Register can be written 1 : Register cannot be written This bit can only be modified if FRD_INTF_CTRL_COMMIT.COMMIT is zero. Reset type: XRSn

9.7.4.7 FRI1_INTF_CTRL_COMMIT Register (Offset = 38h) [Reset = 00000000h]

FRI1_INTF_CTRL_COMMIT is shown in Figure 9-20 and described in Table 9-23.

Return to the Summary Table.

Flash Read Interface 1 Control Commit Register

Figure 9-20. FRI1_INTF_CTRL_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 9-23. FRI1_INTF_CTRL_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the FRD_INTF_CTRL_LOCK register. This bit cannot be cleared, except by reset. 0 : FRD_INTF_CTRL_LOCK is modifiable 1 : FRD_INTF_CTRL_LOCK is committed permanently Reset type: XRSn

9.7.4.8 FRI1_INTF_CLR Register (Offset = 3Ch) [Reset = 00000000h]

FRI1_INTF_CLR is shown in Figure 9-21 and described in Table 9-24.

Return to the Summary Table.

Flash Read Interface 1 Clear Register

Figure 9-21. FRI1_INTF_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					CODE_CACHE_CLR	DATA_CACHE_CLR	PREFETCH_CLR
R-0h					R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 9-24. FRI1_INTF_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	CODE_CACHE_CLR	R-0/W1S	0h	Code cache clear. Self clearing bit that invalidates the data in the code cache when set. Reset type: XRSn
1	DATA_CACHE_CLR	R-0/W1S	0h	Data line buffer clear. Self clearing bit that invalidates the data line buffer data when set. Reset type: XRSn
0	PREFETCH_CLR	R-0/W1S	0h	Prefetch clear. Self clearing bit that invalidates the prefetch buffer data when set. Reset type: XRSn

9.7.4.9 FRI2_INTF_CTRL Register (Offset = 40h) [Reset = 00000000h]

FRI2_INTF_CTRL is shown in Figure 9-22 and described in Table 9-25.

Return to the Summary Table.

Flash Read Interface 2 Control Register

Figure 9-22. FRI2_INTF_CTRL Register

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

Table 9-25. FRI2_INTF_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	RESERVED	R	0h	Reserved
3	DATA_PREREAD_EN	R/W	0h	Sequential data preread mode enable. 0 : A value of 0 disables the mode. 1 : A value of 1 enables the mode. Reset type: XRSn
2	CODE_CACHE_EN	R/W	0h	Code block cache enable. 0 : A value of 0 disables the code cache. 1 : A value of 1 enables the code cache. Reset type: XRSn
1	DATA_CACHE_EN	R/W	0h	Data line buffer enable. 0 : A value of 0 disables the data line buffer. 1 : A value of 1 enables the data line buffer. Reset type: XRSn
0	PREFETCH_EN	R/W	0h	Prefetch enable. 0 : A value of 0 disables prefetch mechanism. 1 : A value of 1 enables prefetch mechanism. Reset type: XRSn

9.7.4.10 FRI2_INTF_CTRL_LOCK Register (Offset = 44h) [Reset = 00000000h]

FRI2_INTF_CTRL_LOCK is shown in Figure 9-23 and described in Table 9-26.

Return to the Summary Table.

Flash Read Interface 2 Control Lock Register

Figure 9-23. FRI2_INTF_CTRL_LOCK Register

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

Table 9-26. FRI2_INTF_CTRL_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-1	RESERVED	R-0	0h	Reserved
0	LOCK	R/W	0h	Determines whether the FRD_INTF_CTRL register can be written. 0 : Register can be written 1 : Register cannot be written This bit can only be modified if FRD_INTF_CTRL_COMMIT.COMMIT is zero. Reset type: XRSn

9.7.4.11 FRI2_INTF_CTRL_COMMIT Register (Offset = 48h) [Reset = 00000000h]

FRI2_INTF_CTRL_COMMIT is shown in Figure 9-24 and described in Table 9-27.

Return to the Summary Table.

Flash Read Interface 2 Control Commit Register

Figure 9-24. FRI2_INTF_CTRL_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 9-27. FRI2_INTF_CTRL_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the FRD_INTF_CTRL_LOCK register. This bit cannot be cleared, except by reset. 0 : FRD_INTF_CTRL_LOCK is modifiable 1 : FRD_INTF_CTRL_LOCK is committed permanently Reset type: XRSn

9.7.4.12 FRI2_INTF_CLR Register (Offset = 4Ch) [Reset = 00000000h]

FRI2_INTF_CLR is shown in Figure 9-25 and described in Table 9-28.

Return to the Summary Table.

Flash Read Interface 2 Clear Register

Figure 9-25. FRI2_INTF_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					CODE_CACHE_CLR	DATA_CACHE_CLR	PREFETCH_CLR
R-0h					R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 9-28. FRI2_INTF_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	CODE_CACHE_CLR	R-0/W1S	0h	Code cache clear. Self clearing bit that invalidates the data in the code cache when set. Reset type: XRSn
1	DATA_CACHE_CLR	R-0/W1S	0h	Data line buffer clear. Self clearing bit that invalidates the data line buffer data when set. Reset type: XRSn
0	PREFETCH_CLR	R-0/W1S	0h	Prefetch clear. Self clearing bit that invalidates the prefetch buffer data when set. Reset type: XRSn

9.7.4.13 FRI3_INTF_CTRL Register (Offset = 50h) [Reset = 00000000h]

FRI3_INTF_CTRL is shown in Figure 9-26 and described in Table 9-29.

Return to the Summary Table.

Flash Read Interface 3 Control Register

Figure 9-26. FRI3_INTF_CTRL Register

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

Table 9-29. FRI3_INTF_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	RESERVED	R	0h	Reserved
3	DATA_PREREAD_EN	R/W	0h	Sequential data preread mode enable. 0 : A value of 0 disables the mode. 1 : A value of 1 enables the mode. Reset type: XRSn
2	CODE_CACHE_EN	R/W	0h	Code block cache enable. 0 : A value of 0 disables the code cache. 1 : A value of 1 enables the code cache. Reset type: XRSn
1	DATA_CACHE_EN	R/W	0h	Data line buffer enable. 0 : A value of 0 disables the data line buffer. 1 : A value of 1 enables the data line buffer. Reset type: XRSn
0	PREFETCH_EN	R/W	0h	Prefetch enable. 0 : A value of 0 disables prefetch mechanism. 1 : A value of 1 enables prefetch mechanism. Reset type: XRSn

9.7.4.14 FRI3_INTF_CTRL_LOCK Register (Offset = 54h) [Reset = 00000000h]

FRI3_INTF_CTRL_LOCK is shown in Figure 9-27 and described in Table 9-30.

Return to the Summary Table.

Flash Read Interface 3 Control Lock Register

Figure 9-27. FRI3_INTF_CTRL_LOCK Register

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

Table 9-30. FRI3_INTF_CTRL_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-1	RESERVED	R-0	0h	Reserved
0	LOCK	R/W	0h	Determines whether the FRD_INTF_CTRL register can be written. 0 : Register can be written 1 : Register cannot be written This bit can only be modified if FRD_INTF_CTRL_COMMIT.COMMIT is zero. Reset type: XRSn

9.7.4.15 FRI3_INTF_CTRL_COMMIT Register (Offset = 58h) [Reset = 00000000h]

FRI3_INTF_CTRL_COMMIT is shown in Figure 9-28 and described in Table 9-31.

Return to the Summary Table.

Flash Read Interface 3 Control Commit Register

Figure 9-28. FRI3_INTF_CTRL_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 9-31. FRI3_INTF_CTRL_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the FRD_INTF_CTRL_LOCK register. This bit cannot be cleared, except by reset. 0 : FRD_INTF_CTRL_LOCK is modifiable 1 : FRD_INTF_CTRL_LOCK is committed permanently Reset type: XRSn

9.7.4.16 FRI3_INTF_CLR Register (Offset = 5Ch) [Reset = 00000000h]

FRI3_INTF_CLR is shown in Figure 9-29 and described in Table 9-32.

Return to the Summary Table.

Flash Read Interface 3 Clear Register

Figure 9-29. FRI3_INTF_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					CODE_CACHE_CLR	DATA_CACHE_CLR	PREFETCH_CLR
R-0h					R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 9-32. FRI3_INTF_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	CODE_CACHE_CLR	R-0/W1S	0h	Code cache clear. Self clearing bit that invalidates the data in the code cache when set. Reset type: XRSn
1	DATA_CACHE_CLR	R-0/W1S	0h	Data line buffer clear. Self clearing bit that invalidates the data line buffer data when set. Reset type: XRSn
0	PREFETCH_CLR	R-0/W1S	0h	Prefetch clear. Self clearing bit that invalidates the prefetch buffer data when set. Reset type: XRSn

9.7.4.17 FRI4_INTF_CTRL Register (Offset = 60h) [Reset = 00000000h]

FRI4_INTF_CTRL is shown in Figure 9-30 and described in Table 9-33.

Return to the Summary Table.

Flash Read Interface 4 Control Register

Figure 9-30. FRI4_INTF_CTRL Register

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

Table 9-33. FRI4_INTF_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	RESERVED	R	0h	Reserved
3	DATA_PREREAD_EN	R/W	0h	Sequential data preread mode enable. 0 : A value of 0 disables the mode. 1 : A value of 1 enables the mode. Reset type: XRSn
2	CODE_CACHE_EN	R/W	0h	Code block cache enable. 0 : A value of 0 disables the code cache. 1 : A value of 1 enables the code cache. Reset type: XRSn
1	DATA_CACHE_EN	R/W	0h	Data line buffer enable. 0 : A value of 0 disables the data line buffer. 1 : A value of 1 enables the data line buffer. Reset type: XRSn
0	PREFETCH_EN	R/W	0h	Prefetch enable. 0 : A value of 0 disables prefetch mechanism. 1 : A value of 1 enables prefetch mechanism. Reset type: XRSn

9.7.4.18 FRI4_INTF_CTRL_LOCK Register (Offset = 64h) [Reset = 00000000h]

FRI4_INTF_CTRL_LOCK is shown in Figure 9-31 and described in Table 9-34.

Return to the Summary Table.

Flash Read Interface 4 Control Lock Register

Figure 9-31. FRI4_INTF_CTRL_LOCK Register

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

Table 9-34. FRI4_INTF_CTRL_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-1	RESERVED	R-0	0h	Reserved
0	LOCK	R/W	0h	Determines whether the FRD_INTF_CTRL register can be written. 0 : Register can be written 1 : Register cannot be written This bit can only be modified if FRD_INTF_CTRL_COMMIT.COMMIT is zero. Reset type: XRSn

9.7.4.19 FRI4_INTF_CTRL_COMMIT Register (Offset = 68h) [Reset = 00000000h]

FRI4_INTF_CTRL_COMMIT is shown in Figure 9-32 and described in Table 9-35.

Return to the Summary Table.

Flash Read Interface 4 Control Commit Register

Figure 9-32. FRI4_INTF_CTRL_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 9-35. FRI4_INTF_CTRL_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the FRD_INTF_CTRL_LOCK register. This bit cannot be cleared, except by reset. 0 : FRD_INTF_CTRL_LOCK is modifiable 1 : FRD_INTF_CTRL_LOCK is committed permanently Reset type: XRSn

9.7.4.20 FRI4_INTF_CLR Register (Offset = 6Ch) [Reset = 00000000h]

FRI4_INTF_CLR is shown in Figure 9-33 and described in Table 9-36.

Return to the Summary Table.

Flash Read Interface 4 Clear Register

Figure 9-33. FRI4_INTF_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					CODE_CACHE_CLR	DATA_CACHE_CLR	PREFETCH_CLR
R-0h					R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 9-36. FRI4_INTF_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	CODE_CACHE_CLR	R-0/W1S	0h	Code cache clear. Self clearing bit that invalidates the data in the code cache when set. Reset type: XRSn
1	DATA_CACHE_CLR	R-0/W1S	0h	Data line buffer clear. Self clearing bit that invalidates the data line buffer data when set. Reset type: XRSn
0	PREFETCH_CLR	R-0/W1S	0h	Prefetch clear. Self clearing bit that invalidates the prefetch buffer data when set. Reset type: XRSn

9.7.4.21 PARITY_TEST Register (Offset = 70h) [Reset = 00000000h]

PARITY_TEST is shown in Figure 9-34 and described in Table 9-37.

Return to the Summary Table.

Parity Test Enable

Figure 9-34. PARITY_TEST Register

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

Table 9-37. PARITY_TEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	RESERVED	R	0h	Reserved
3-0	TESTEN	R/W	0h	Configures the parity test feature. 1010 : Parity test feature is enabled Other : Parity test feature is disabled When a parity protected FRI register is read while parity test is enabled, the least significant bit in each byte indicates the parity test result for that byte. 0 : No parity error 1 : Parity error When a parity protected FRI register byte is written with a '1' in the least significant bit of a byte while parity test is enabled, that byte's stored parity bit's value is inverted, thereby injecting an error. A write of '0' to these bits has no effect. It is recommended to leave the field as 0101 or 0000 after completing the parity test. Reset type: XRSn

9.7.4.22 PARITY_TEST_LOCK Register (Offset = 74h) [Reset = 00000000h]

PARITY_TEST_LOCK is shown in Figure 9-35 and described in Table 9-38.

Return to the Summary Table.

Parity Test Lock Register

Figure 9-35. PARITY_TEST_LOCK Register

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

Table 9-38. PARITY_TEST_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-1	RESERVED	R-0	0h	Reserved
0	LOCK	R/W	0h	Determines whether the PARITY_TEST register can be written. 0 : Register can be written 1 : Register cannot be written This bit can only be modified if PARITY_TEST_COMMIT.COMMIT is zero. Reset type: XRSn

9.7.4.23 PARITY_TEST_COMMIT Register (Offset = 78h) [Reset = 00000000h]

PARITY_TEST_COMMIT is shown in Figure 9-36 and described in Table 9-39.

Return to the Summary Table.

Parity Test Commit Register

Figure 9-36. PARITY_TEST_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 9-39. PARITY_TEST_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the PARITY_TEST_LOCK register. This bit cannot be cleared, except by reset. 0 : PARITY_TEST_LOCK is modifiable 1 : PARITY_TEST_LOCK is committed permanently Reset type: XRSn

Chapter 10

Safety and Security Unit (SSU)

This chapter describes the features and operation of the Safety and Security Unit (SSU). The SSU is the central element of the C29x security architecture. The SSU provides run-time safety and security for the application CPU subsystems, and governs debug access to various resources on the chip. The SSU also provides hardware infrastructure for managing Flash firmware update operations, using the Firmware-Over-The-Air (FOTA) or Live Firmware Update (LFU) processes.

10.1 Introduction

At the heart of the architecture is the Safety and Security Unit (SSU). The SSU acts as a firewall between the C29x CPUs and the memory and peripherals. The primary role of the SSU is to enforce user access protection policy every time the C29x CPU performs accesses to peripherals and memory on the chip. In addition, the SSU governs debug access and Flash Controller operations in the C29x application subsystem (note: the SSU has no control over the HSM Flash, or any other HSM resources). While the Hardware Security Module (HSM) provides cryptographic services and governs authentication, secure boot and secure key/code provisioning, the SSU is responsible for run-time safety and security protections in application CPU subsystems. Both the HSM and SSU govern debug access authorization; both must enable access to a specific resource for debug to be authorized. For more information on debug access authorization, see Section 10.11.

10.1.1 SSU Related Collateral

Foundational Materials

- C29x Academy - Safety Security Unit (SSU)

Expert Materials

- *Implementing Run-Time Safety and Security Protections with SSU Application Report*

10.1.2 Block Diagram

A simplified view of the F29x Real-Time Security architecture in this device is shown in Figure 10-1.

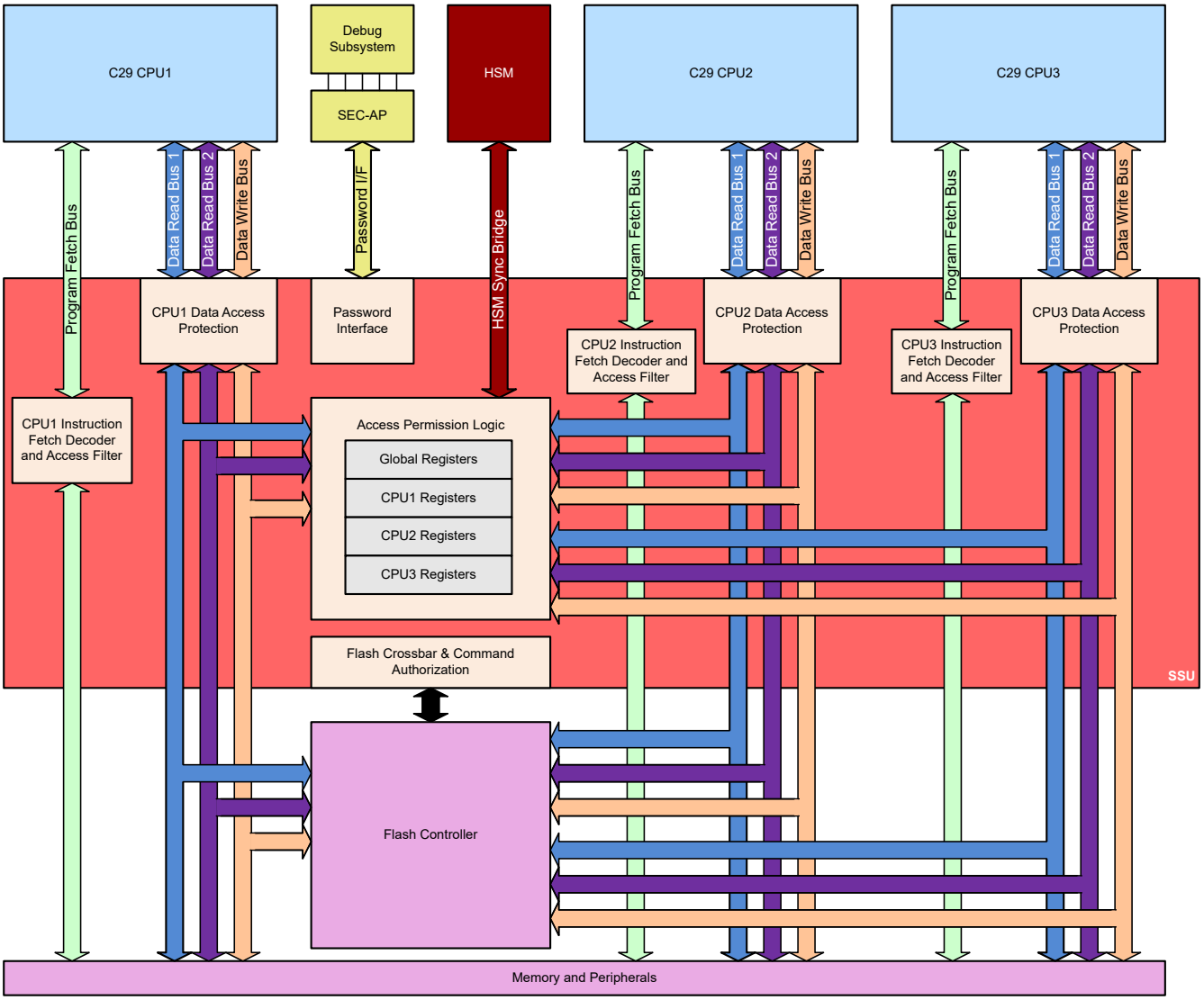


Figure 10-1. C29x Real-Time Security Architectural Block Diagram

10.1.3 System SSU Configuration Example

The SSU is tightly coupled to the C29x CPUs and the C29x Flash Controllers. Each C29x CPU is designed to support hardware function isolation and protections using memory protection identifiers (LINKs), safety and security isolation contexts (STACKs), and debug access ZONEs. An example of a system SSU configuration, showing the relationship between access protection ranges, LINKs, STACKs and ZONEs is shown in Figure 10-2. When the CPU requests an instruction fetch, the SSU first decodes the instruction address to a LINK, STACK, and ZONE, and then passes that information back to the CPU along with the fetched data. The CPU retains this security context information together with the instruction throughout the execution pipeline, and passes the context along to the SSU when making a data memory read or write access.

Each LINK consists of one or more regions of executable code, defined by the memory ranges in which the code resides, and is typically associated with a specific task or software module. The Access Protection Ranges, or APRs, are the basic unit of memory protection. Each APR has a start address and an end address, and can be associated with executable code (associating that code with a specific LINK), or can cover data or peripheral memory. Each APR then defines read and write access permissions for every LINK available on the CPU, enabling dynamic memory access permissions that automatically change based on which code is accessing the memory region. Each LINK is associated with one STACK. Each STACK is associated with a physically distinct and separate stack pointer in the CPU for secure code isolation. Finally, every STACK is associated with one device-level ZONE that governs debug and firmware update security.

These concepts are explained in detail in the following sections.

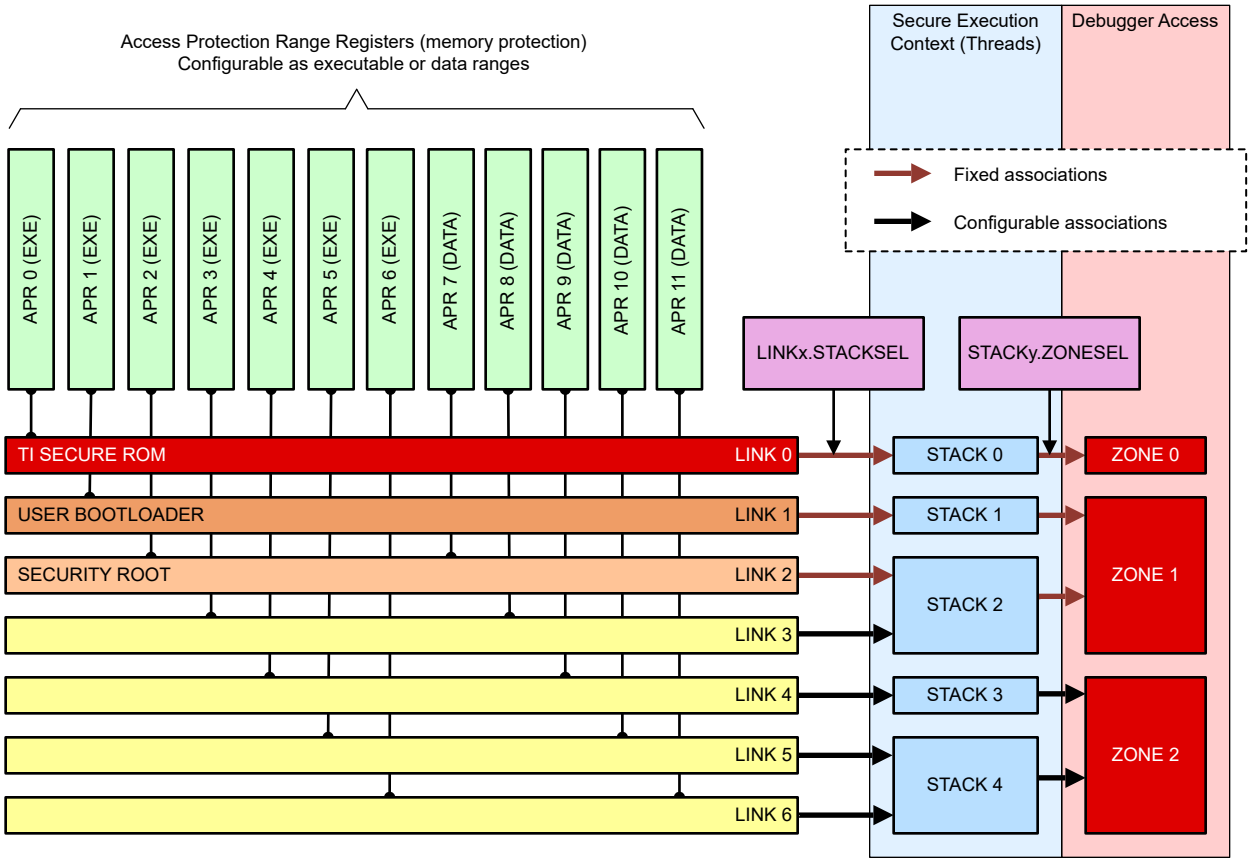


Figure 10-2. System SSU Configuration Example

10.2 Access Protection Ranges

The SSU defines memory protections using Access Protection Ranges (APRs), governing CPU access to on-chip memories, peripherals, and external memories. There are a total of 64 APRs available per CPU. Each APR specifies the following:

- Starting address
- Ending address
- Associated code execution LINK
- Code execution permissions
- Read and write permissions per LINK
- Inherited permissions
- Lock and commit settings

Access protection ranges are defined using the APx registers. The access protection registers are loaded at boot time, but can be modified afterward by code running with security root privileges. Before performing a CPU access to any memory location, the user must make sure to define an access protection range that includes that location. In addition, access protection range definitions must not overlap each other. If a CPU access is performed to a memory location that is covered by zero or more than one access protection range, the SSU generates a fault signal to the CPU, which then forwards the error to the Error Aggregator module.

Each standard AP range has a start address, defined in the APx_START register, and an end address, defined in the APx_END register. Every AP range is aligned to a 4KB memory boundary, so the lower 12 bits of each APx_START register are hardcoded to 0, and the lower 12 bits of each APx_END register are hardcoded to 0xFFF. The regular AP ranges can address the first 2GB of the device address space, covering on-chip memories or peripheral memory-mapped registers. Each AP range has a maximum size of 16MB. The MEMTYPE field in the start address indicates what type of memory is covered by the AP range. Note that the MEMTYPE bits are the same as the CPU address bits per the device memory-map; there is no special handling of these bits in the SSU. Simply writing the range's start address to APx_START is sufficient.

Access permissions for an AP range are configured using the APx_ACCESS register. This register defines, for each LINK, whether to grant no access permission, read-only permission, or read/write permission.

AP ranges also include the following configuration options:

- Access Protection Disable, by writing to the APx_CFG.APD bit.
- Execute Enable/Disable, by writing to the APx_CFG.XE bit. This setting defines the AP region as a code/program memory region, and any code instruction that is fetched from the memory range is associated with the LINK defined in the APx_CFG.LINKID field.

The first two AP ranges (AP0 and AP1) are extended AP ranges. These special ranges have address ranges spanning the full 4GB address space available, and as such can cover extended memory ranges such as external memories connected using the EMIF interface. The start and end addresses of these extended AP ranges are defined in the APx_START_EXT and APx_END_EXT registers. Like standard AP ranges, the extended AP ranges must also be aligned to 4KB address boundaries.

Note

Not all memory regions can be configured for protection by an AP range. Certain peripherals, memory regions and bridges are governed by system-level hardcoded protections. For a detailed memory map showing which memory regions can be APR-protected, see the device data sheet.

10.2.1 Access Protection Inheritance

An AP range can be configured to inherit protection settings from a specific LINK's caller. The APx_CFG.APILINK register field specifies the LINKID which inherits permissions for access to Access Protection Range APx. The C29x CPU sets DSTS.CLINK to the current LINK's caller LINKID whenever a protected call is performed. Whenever code belonging to the LINKn specified in APx_CFG.APILINK performs an access to memory within APx, the access permissions granted to LINKn match the access permissions for LINKn's caller (as identified by CLINK on the data access request). The access protection inheritance feature is enabled by setting APx_CFG.APILINKE to 1.

Access protection inheritance is useful for maintaining memory protection when using common or utility functions such as *memcpy*. Without inheritance, an unauthorized LINK can access data memory covered by APx by simply calling a shared memory function with direct read or write access to APx. However, when APx_CFG.APILINKE is set to 1 and APx_CFG.APILINK is defined to be the LINK associated with the common code functions, then the shared function is imputed the same permissions as the LINK that called the shared function. An illustration of how access protection inheritance enables the sharing of common code functions between different LINKs is shown in Figure 10-3.

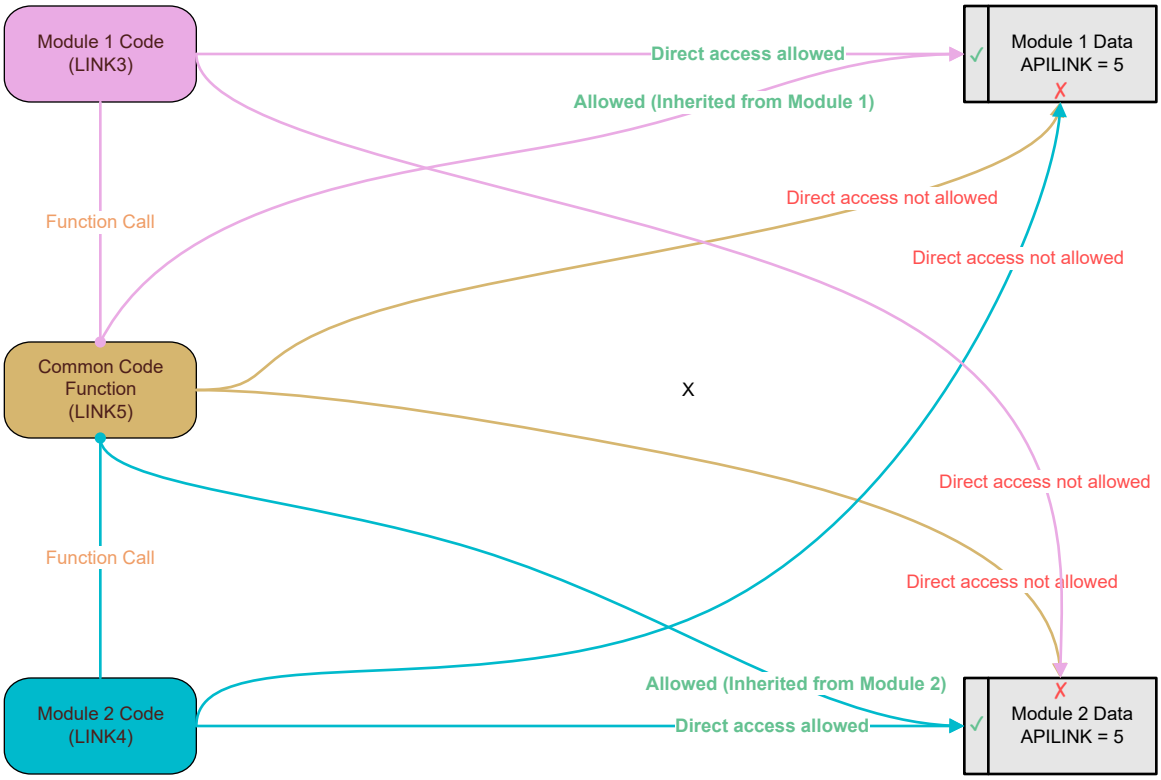


Figure 10-3. Access Protection Inheritance Example

Note

In the C29x CPU, DSTS.CLINK is only updated by a protected function call, and is not updated during a regular function call or branch. For LINKs within the same STACK, branches and function calls must be executed using protected call instructions for the access protection inheritance feature to work correctly.

10.3 LINKs

LINKs form the basis for context-sensitive memory protection. Each LINK represents one or more regions of executable code, and determines what data memory ranges (APRs) can be accessed by that code. Every code instruction that executes on the CPU has an associated LINK ID, which is assigned based on the memory region the instruction was fetched from. Thus, memory protections across the entire SoC can be automatically configured for each instruction at execution time, eliminating the need for software-based MPU reconfiguration.

The APx_CFG.LINKID register defines the execution LINK for all code that the CPU fetches from the APx memory address range. In addition, to enable code execution from an AP range, the user must configure the APx_CFG.XE (execution enable) bit. For data-only LINKs, make sure to set the XE bit to zero to preserve code safety. The SSU inputs the LINKID to the CPU when an instruction is fetched, and the CPU outputs the LINKID to the SSU whenever the CPU performs a data access to a memory location. The SSU compares the requesting LINKID to the access permissions for the APR containing the target memory address by examining the corresponding APx_ACCESS.LINKy register. If the permissions defined in the APx_ACCESS.LINKy do not authorize the requesting LINKID, then the SSU generates a fault to the CPU. If the memory region containing the address is governed by hardcoded protections (e.g. writable by LINK0, LINK1 or LINK2 only), the SSU checks the requested operation against the hardcoded protection, and generates a fault to the CPU if unauthorized. See Figure 10-4 for an illustration of how the SSU filters data access requests from the CPU using associated LINKIDs.

There are 16 LINKs per CPU. The SSU predefines and reserves the following special roles for the first three links:

- LINK0 is reserved for internal use and cannot be configured by the user.
- LINK1 is primarily used for TI and user boot loaders, but can also be used for user run-time code.
- LINK2 is the primary user LINK. CPU1.LINK2 is the Security Root LINK (SROOT).

All other LINKs are secondary user LINKs, and can be defined by the user application as required. LINK0, LINK1 and LINK2 have special hard-coded permissions. For more info, see Section 10.12.

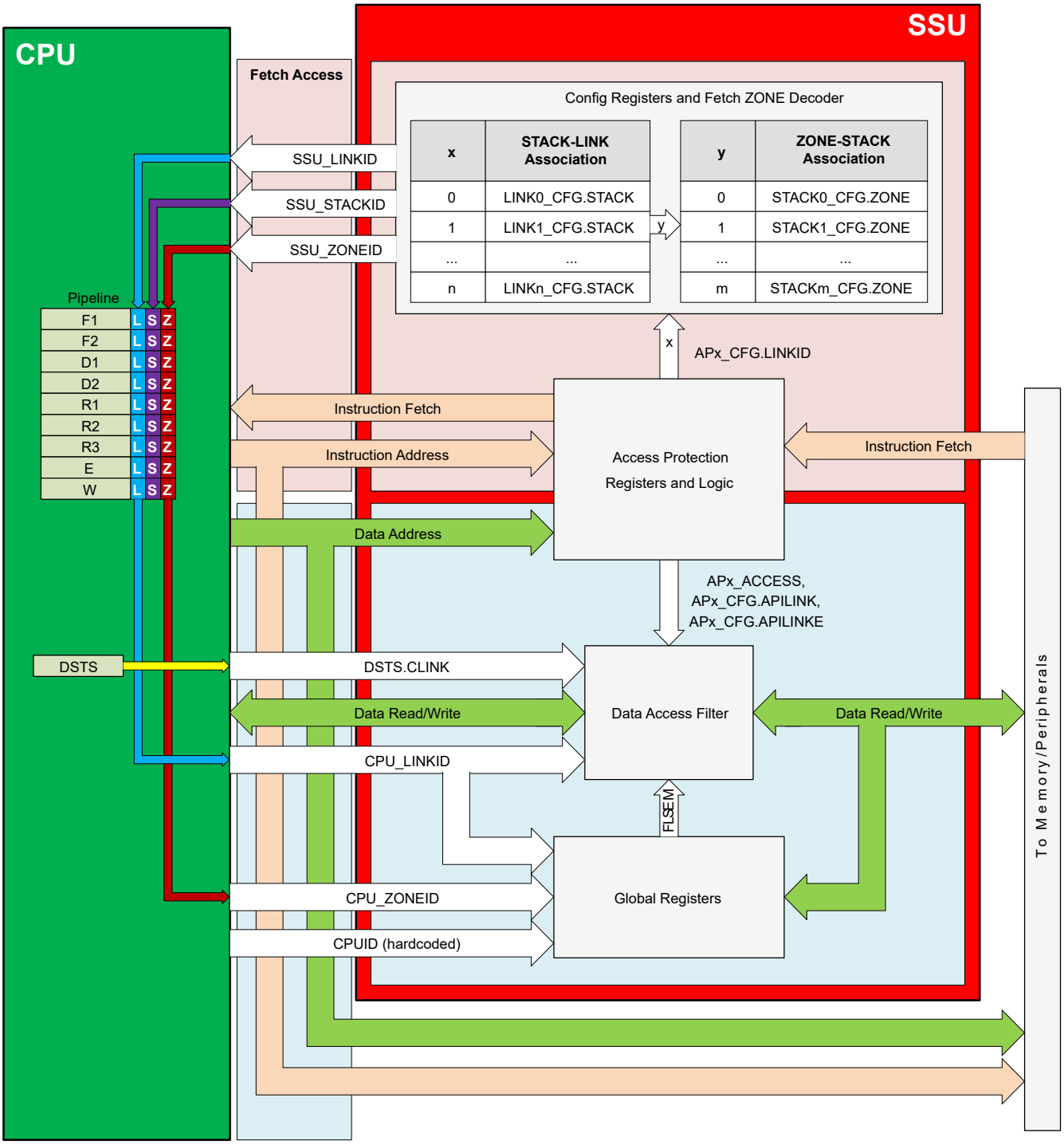


Figure 10-4. SSU-CPU Tightly Coupled Interface

10.4 STACKs

STACKs isolate code execution contexts from each other. Each STACK has a dedicated stack pointer in the CPU, and a dedicated set of CPU registers, and provides hardware safety and security isolation of code from other STACKs. Every LINK belongs to one and only one STACK, but a STACK can contain multiple LINKs. For CPU execution to cross from a LINK associated with one STACK to a LINK associated with another STACK, the code must execute special instructions that preserve data safety during context switching.

To specify which STACK a LINK belongs to, configure the LINKx_CFG field/register in the SSU configuration.

Like LINKs, the SSU passes the STACKID to the CPU during instruction fetch, and the CPU outputs the instruction STACKID to the SSU when performing data accesses.

Each CPU has a total of 8 STACKs. The SSU predefines and reserves the first three stacks as follows:

- STACK0 is reserved for internal use, and is always associated with LINK0 only.
- STACK1 is primarily used for boot loaders, but can also be used for user application code. STACK1 is always associated with LINK1 only.
- STACK2 is the primary user stack. STACK2 is associated with LINK2, and can be associated with other secondary user LINKs depending on the configured user protection policy.

STACK3 and above are secondary user STACKs, and can be defined by the user application as required.

10.5 ZONEs

Each ZONE governs debugger access to the system, and as well as firmware update permissions. ZONEs determine what permissions an external user or programmer has to download or program code, perform firmware updates, and debug the system. While AP ranges, LINKs and STACKs are independently defined per CPU, ZONEs span the entire device, excluding the HSM (which is not governed by the SSU).

Each device has a total of 4 available ZONEs. ZONE0 is reserved for TI internal use, and is always associated with STACK0/LINK0. The primary user ZONE is ZONE1. ZONE1 is always associated with STACK1 and STACK2, but can include other STACKs depending on user configuration. ZONE2 and ZONE3 are secondary user ZONEs, and can be associated with STACK3 and other secondary user STACKs.

To configure which ZONE a STACK is associated with, write to the STACKx_CFG field/register in the SECCFG sector.

Each user ZONE has three possible operating permission levels:

- **Debug not allowed**
- **Partial debug allowed:** This mode permits the user to debug CPU execution (halt, resume, view registers), but prevents read/write access to memories.
- **Full debug allowed:** This mode enables full debug capability.

In partial debug mode, the SSU blocks debugger access to various registers, as described in Section 10.13. For information on how to enable debug for user ZONEs, see Section 10.11. The SSU_GEN_REGS.DEBUG_STAT register indicates the current active debug permission level for each user ZONE.

A debug ZONE inherits memory access permissions from the sum total of all LINKs that are associated with the ZONE. This means that if a ZONE is enabled for full debug, then the debugger's ability to read or write any memory address depends on whether the access protection range covering that address gives permission to any LINK belonging to the ZONE.

Note

If an AP range has the APx_CFG.XE bit set (that is, defined as a code region), and the LINKID of the AP region belongs to a ZONE that is enabled for full debug, then the debugger is given read and write access to the AP range—even if no associated LINKs have read/write access to the AP range.

10.6 SSU-CPU Interface

The SSU is tightly coupled with each CPU during different phases of execution. When the CPU requests an instruction fetch, the SSU first decodes the instruction address to a LINK, STACK, and ZONE, and passes that information back to the CPU along with the fetched data. The CPU retains this security context information together with the instruction throughout the execution pipeline, and passes the context along to the SSU when making a data memory read or write access. Within the SSU, the data access filter logic uses the LINKID and CLINK associated with the instruction to qualify each memory access request and determine whether to permit the access.

The SSU provides a CPUID register, can be read by the application to determine which CPU the application is executing on. Additionally, the decoder logic can be used to determine which LINK, STACK and ZONE a particular code instruction belongs to, by writing the instruction's address to the DECODER_ADDR_IN register and reading the DECODER_OUT register.

Figure 10-4 illustrates the tightly-coupled interface between the SSU and the C29x CPU.

10.6.1 SSU Operation in Lockstep Mode

When lockstep mode is enabled on a CPU pair (for example, CPU1 and CPU2), the SSU enforces protections based on the security policy for CPU1. Security sideband signals for LINK, STACK, ZONE and protection fault status are forwarded to both CPUs in lockstep. For a description of functional safety requirement compliance in this mode, see the device safety manual.

Note

Flash programming and erase operations cannot be performed in lockstep. To implement a functional safety strategy for Flash programming, incorporate verify reads of the programmed Flash content after programming. For more information, see the device safety manual.

10.7 SSU Operation Modes

The SSU has three modes of operation: SSUMODE1, SSUMODE2, and SSUMODE3. Devices ship from the TI factory in SSUMODE1 mode. Once safety and security has been integrated into the development project, configure the SSU to SSUMODE2 to enable safety/code isolation features, and then configure to SSUMODE3 when development is finalized for production programming to enable full security. If safety and security features are not desired in the target application, the device can be left in SSUMODE1; and if safety features are desired but not security, the device can be left in SSUMODE2.

In SSUMODE3:

1. All safety and security features of the SSU are active and enforced.
2. All user ZONEs (ZONE1, ZONE2, and ZONE3) are disabled for debug by default. Each ZONE must be authorized as configured to enable partial or full debug access.
3. JTAG and debug access are governed by debug authorization settings.
4. Debugger accesses to LINK1 RAM and LINK1 Flash are only permitted when ZONE1 is enabled for full debug.

In SSUMODE2:

1. JTAG is enabled;
2. All user ZONEs (ZONE1, ZONE2, and ZONE3) are enabled for full debug;
3. CPU1.LINK1 and/or any C29x CPU debug access is allowed to:
   a. Write to the RAMOPENFRC and RAMOPENCLR registers
   b. Write to the SSU APx registers
   c. Update all Flash SECCFG sectors
   d. Update all Flash MAIN sectors

In SSUMODE1:

1. All characteristics of SSUMODE2 are present;
2. User access protection (AP) regions are disabled for all CPUs;
3. The entire memory map of the device, excluding ROM, RAMOPENed RAM and hard-coded RAM and Flash memory regions, is mapped to run as LINK2;
4. The entire memory map of the device, excluding ROM, RAMOPENed RAM and hardcoded RAM and Flash memory regions, is accessible by all LINKs.

Both SSUMODE1 and SSUMODE2 permit generic programmers to program Flash memory. However, in SSUMODE3, the user must explicitly give LINK1 Flash memory write permissions to enable any kind of bootloader/Flash kernel to program the Flash.

SSUMODE must be configured by programming SSU_MODE location in Section 10.8.2. The value of SSUMODE is loaded from SECCFG into the SSUMODE register during boot. The value of SSUMODE cannot be changed during run time.

Note

SSUMODE is a 32-bit register. All 32 bits must be programmed into the SECCFG field, including leading zeroes, for the device to boot successfully.

Although all user ZONEs are enabled for full debug in SSUMODE2, this does not automatically give the debugger full access to device memory. Every debug ZONE inherits permissions from the LINKs belonging to that ZONE, and these permissions are configured in the AP ranges. If there is no AP range covering a memory address, debugger accesses to that memory address are blocked in SSUMODE2. To enable debugger access to a memory region in SSUMODE2, make sure to configure an AP range covering the region, and grant at least one LINK the access level required for the debugger to access the memory region.

10.8 Security Configuration and Flash Management

Each Flash bank in the C29x application subsystem has two regions that contain information which configures the SSU: SECCFG and BANKMGMT. These regions are separate from the main Flash sectors, and do not count against the stated available Flash memory for application use. SECCFG is used for storing the SSU configuration, also known as the User Protection Policy (UPP). BANKMGMT is used for managing Flash updates using the Firmware-Over-The-Air (FOTA) or Live Firmware Update (LFU) mechanisms.

The configuration settings stored in these Flash regions are loaded into the SSU registers at device startup. Most UPP settings cannot be modified by the application during run time. To modify the UPP, the application or connected debugger must erase and re-program SECCFG with the desired settings. See Section 10.9 for details on protection mechanisms to prevent unauthorized updates to these sectors. Section 10.13 contains details on run-time access permissions for individual SSU registers.

Texas Instruments recommends using the SysConfig tool, included as part of the device MCU-SDK, to configure the SSU and generate user security settings. This tool provides an easy-to-use graphical user interface to create application security partitions and protection definitions, and automatically generates linker command files, a SECCFG image, and header files to be included in the user application.

Note

An accurate SECCFG CRC value is required for the device to boot successfully. To avoid device boot problems due to inaccurate CRC computation, use the SysConfig tool to build the SECCFG image and auto-compute the associated CRC.

10.8.1 BANKMGMT Sectors

The BANKMGMT region of each bank pair contains three fields that are used to manage bank status and firmware updates:

1. **BANK_STATUS[63:0]** (Offset 0x00): This field specifies whether the bank contents are currently valid or invalid. A valid bank pair has a status value of 0x55555555_55555555. If both banks are inactive, the device is unable to execute Flash code and can fail to boot. If both banks have valid active statuses, then the current active bank is determined by BANK_UPDATE_CTR.
2. **BANK_UPDATE_CTR[63:0]** (Offset 0x08): This is a 64-bit *decrementing* counter value that is used to determine the most recent firmware revision. The counter has a value of 0xFFFFFFFF_FFFFFFFF when shipped from the TI factory. When both bank pairs have valid status values, the BANK_UPDATE_CTR for each bank pair is compared against the other to determine the most recent firmware revision. The bank pair with the lower counter value is determined to be the most recent revision.
3. **BANKMODE[63:0]** (Offset 0x10): This field only exists in FLC1 BANKMGMT regions (for CPU1). The BANKMODE field configures the device bankmode and is loaded into the SSU_GEN_REGS.BANKMODE register at device boot. After BANKMODE is configured, the boot ROM then reads the value of each bank pair's BANK_UPDATE_CTR to determine whether CPUxSWAP is enabled.

The BANKMGMT region is best programmed and configured using the Flash API and TI Flash tools, such as the CCS Flash Plugin or UniFlash. To avoid misconfiguration resulting in an unbootable device, do not allocate data directly into the BANKMGMT region in the application image/.out file.

The following security restrictions apply to updates of the BANKMGMT region:

- In SSUMODE1, all code running from any LINK has permission to update the BANKMGMT sectors, since ZONE1 is enabled for full debug. This enables ease of code development.
- In SSUMODE2, all code running from any LINK has permission to update the BANKMGMT sectors, since ZONE1 is enabled for full debug.
- In SSUMODE3, only the SECCFG update owner defined in SECCFG_UPDATE_CFG has permission to update BANKMGMT sectors. In addition, SECCFG_UPDATE_CFG.UPDATE_EN must be set to 0xC to enable updates to BANKMGMT when operating in SSUMODE3.

10.8.2 SECCFG Sectors

SECCFG sectors are a portion of Flash memory designated for storing the SSU user configuration, or User Protection Policy (UPP). The following rules apply to SECCFG sectors:

- Each SECCFG sector is 2KB in size.
- Each CPU pair (for example, CPU1/CPU2) has an associated active SECCFG sector, and a reserve SECCFG sector.
- All code banks are interleaved in pairs to enable 256-bit instruction fetches. As a result, the effective SECCFG sector size is 4KB, due to the interleaving of the two banks. However, the SECCFG image size is 2KB; the remaining half is unused.

Table 10-1 shows the mapping of SECCFG sectors to CPU addresses. Each CPU has an active SECCFG start address, and an alternate (reserve) SECCFG start address. Reserve SECCFG sectors are used for Flash updates, and are swapped with the active sector (together with all other bank sectors) when the firmware update process is complete. The locations of these addresses in the CPU memory map depend on the active settings in the BANKMODE and BANKMAP registers. The SECVALID register, in combination with the BANKMODE register, defines which address holds the valid SECCFG information for a CPU pair. The CPUxSWAP fields in the BANKMAP register specify which bank contains this information; however, the address range does not change even if CPUxSWAP changes.

The values of BANKMODE and BANKMAP in the Flash BANKMGMT sectors are automatically loaded into the SSU registers at boot time. After determining the active SECCFG sector, the device then loads the User Protection Policy from the active SECCFG sector into the SSU registers.

Table 10-2 shows a map of the SECCFG sector contents as mapped to SSU registers.

10.8.3 SECCFG Sector Address Mapping

Table 10-1 shows the mapping of physical bank SECCFG sectors to CPU addresses, depending on the Flash bank mode and swap configuration.

Table 10-1. SECCFG Address Mapping

Bank Mode	CPUxSWAP	Region	CPU1/CPU2 Bank	CPU1 Address	CPU2 Address	CPU3 Bank	CPU3 Address
MODE 0		Active	FLC1.B0/B1	0x10D8 1000	0x10D8 1800	FLC2.B0/B1	0x10D8 9000
		Update	FLC1.B2/B3	0x10D8 5000	0x10D8 5800	FLC2.B2/B3	0x10D8 D000
MODE 1	SWAP = 0	Active	FLC1.B0/B1	0x10D8 1000	0x10D8 1800	FLC2.B0/B1	0x10D8 5000
		Update	FLC1.B2/B3	0x10D9 9000	0x10D9 9800	FLC2.B2/B3	0x10D9 D000
	SWAP = 1	Active	FLC1.B2/B3	0x10D8 1000	0x10D8 1800	FLC2.B2/B3	0x10D8 5000
		Update	FLC1.B0/B1	0x10D9 9000	0x10D9 9800	FLC2.B0/B1	0x10D9 D000
MODE 2		Active	FLC1.B0/B1	0x10D8 1000	0x10D8 1800	FLC2.B0/B1	0x10D9 1000
		Update	FLC1.B2/B3	0x10D8 5000	0x10D8 5800	FLC2.B2/B3	0x10D9 5000
MODE 3	SWAP = 0	Active	FLC1.B0/B1	0x10D8 1000	0x10D8 1800	FLC2.B0/B1	0x10D9 1000
		Update	FLC1.B2/B3	0x10D9 9000	0x10D9 9800	FLC2.B2/B3	0x10D9 D000
	SWAP = 1	Active	FLC1.B2/B3	0x10D8 1000	0x10D8 1800	FLC2.B2/B3	0x10D9 1000
		Update	FLC1.B0/B1	0x10D9 9000	0x10D9 9800	FLC2.B0/B1	0x10D9 D000

10.8.4 SECCFG Sector Memory Map

Table 10-2 shows the mapping of various SSU registers and configuration values to SECCFG sector address locations.

Table 10-2. SECCFG Sector Memory Map

Section	Address Offset	CPUs	Bits 31:24	Bits 23:16	Bits 15:8	Bits 7:0
Access Protection Configurations	0x000	All CPUs	AP0_START[31:12]; Reserved[11:1]; AP0_LOCK[0]			
	0x004	All CPUs	AP0_END[31:12]/Reserved[11:1]/AP0_COMMIT[0]			
	0x008	All CPUs	AP0_ACCESS			
	0x00C	All CPUs	AP0_CFG[15:0]		Reserved	
	0x010	All CPUs	AP1_START[31:12]; Reserved[11:1]; AP1_LOCK[0]			
	0x014	All CPUs	AP1_END[31:12]/Reserved[11:1]/AP1_COMMIT[0]			
	0x018	All CPUs	AP1_ACCESS			
	0x01C	All CPUs	AP1_CFG[15:0]		Reserved	
	0x020	All CPUs	AP2_START[31:12]; Reserved[11:1]; AP2_LOCK[0]			
	0x024	All CPUs	AP2_END[31:12]/Reserved[11:1]/AP2_COMMIT[0]			
	0x028	All CPUs	AP2_ACCESS			
	0x02C	All CPUs	AP2_CFG[15:0]		Reserved	
	0x030	All CPUs	AP3_START[31:12]; Reserved[11:1]; AP3_LOCK[0]			
	0x034	All CPUs	AP3_END[31:12]/Reserved[11:1]/AP3_COMMIT[0]			
	0x038	All CPUs	AP3_ACCESS			
	0x03C	All CPUs	AP3_CFG[15:0]		Reserved	
	0x040	All CPUs	AP4_START[31:12]; Reserved[11:1]; AP4_LOCK[0]			
	0x044	All CPUs	AP4_END[31:12]/Reserved[11:1]/AP4_COMMIT[0]			
	0x048	All CPUs	AP4_ACCESS			
	0x04C	All CPUs	AP4_CFG[15:0]		Reserved	
	0x050	All CPUs	AP5_START[31:12]; Reserved[11:1]; AP5_LOCK[0]			
	0x054	All CPUs	AP5_END[31:12]/Reserved[11:1]/AP5_COMMIT[0]			
	0x058	All CPUs	AP5_ACCESS			
	0x05C	All CPUs	AP5_CFG[15:0]		Reserved	
	0x060	All CPUs	AP6_START[31:12]; Reserved[11:1]; AP6_LOCK[0]			
	0x064	All CPUs	AP6_END[31:12]/Reserved[11:1]/AP6_COMMIT[0]			
	0x068	All CPUs	AP6_ACCESS			
	0x06C	All CPUs	AP6_CFG[15:0]		Reserved	
	0x070	All CPUs	AP7_START[31:12]; Reserved[11:1]; AP7_LOCK[0]			
	0x074	All CPUs	AP7_END[31:12]/Reserved[11:1]/AP7_COMMIT[0]			
	0x078	All CPUs	AP7_ACCESS			
	0x07C	All CPUs	AP7_CFG[15:0]		Reserved	
	0x080	All CPUs	AP8_START[31:12]; Reserved[11:1]; AP8_LOCK[0]			
	0x084	All CPUs	AP8_END[31:12]/Reserved[11:1]/AP8_COMMIT[0]			
	0x088	All CPUs	AP8_ACCESS			
	0x08C	All CPUs	AP8_CFG[15:0]		Reserved	
	0x090	All CPUs	AP9_START[31:12]; Reserved[11:1]; AP9_LOCK[0]			
	0x094	All CPUs	AP9_END[31:12]/Reserved[11:1]/AP9_COMMIT[0]			
	0x098	All CPUs	AP9_ACCESS			
	0x09C	All CPUs	AP9_CFG[15:0]		Reserved	
	0x0A0	All CPUs	AP10_START[31:12]; Reserved[11:1]; AP10_LOCK[0]			

Table 10-2. SECCFG Sector Memory Map (continued)

Section	Address Offset	CPUs	Bits 31:24	Bits 23:16	Bits 15:8	Bits 7:0
	0x0A4	All CPUs	AP10_END[31:12]/Reserved[11:1]/AP10_COMMIT[0]			
	0x0A8	All CPUs	AP10_ACCESS			
	0x0AC	All CPUs	AP10_CFG[15:0]		Reserved	
	0x0B0	All CPUs	AP11_START[31:12]; Reserved[11:1]; AP11_LOCK[0]			
	0x0B4	All CPUs	AP11_END[31:12]/Reserved[11:1]/AP11_COMMIT[0]			
	0x0B8	All CPUs	AP11_ACCESS			
	0x0BC	All CPUs	AP11_CFG[15:0]		Reserved	
	0x0C0	All CPUs	AP12_START[31:12]; Reserved[11:1]; AP12_LOCK[0]			
	0x0C4	All CPUs	AP12_END[31:12]/Reserved[11:1]/AP12_COMMIT[0]			
	0x0C8	All CPUs	AP12_ACCESS			
	0x0CC	All CPUs	AP12_CFG[15:0]		Reserved	
	0x0D0	All CPUs	AP13_START[31:12]; Reserved[11:1]; AP13_LOCK[0]			
	0x0D4	All CPUs	AP13_END[31:12]/Reserved[11:1]/AP13_COMMIT[0]			
	0x0D8	All CPUs	AP13_ACCESS			
	0x0DC	All CPUs	AP13_CFG[15:0]		Reserved	
	0x0E0	All CPUs	AP14_START[31:12]; Reserved[11:1]; AP14_LOCK[0]			
	0x0E4	All CPUs	AP14_END[31:12]/Reserved[11:1]/AP14_COMMIT[0]			
	0x0E8	All CPUs	AP14_ACCESS			
	0x0EC	All CPUs	AP14_CFG[15:0]		Reserved	
	0x0F0	All CPUs	AP15_START[31:12]; Reserved[11:1]; AP15_LOCK[0]			
	0x0F4	All CPUs	AP15_END[31:12]/Reserved[11:1]/AP15_COMMIT[0]			
	0x0F8	All CPUs	AP15_ACCESS			
	0x0FC	All CPUs	AP15_CFG[15:0]		Reserved	
	0x100	All CPUs	AP16_START[31:12]; Reserved[11:1]; AP16_LOCK[0]			
	0x104	All CPUs	AP16_END[31:12]/Reserved[11:1]/AP16_COMMIT[0]			
	0x108	All CPUs	AP16_ACCESS			
	0x10C	All CPUs	AP16_CFG[15:0]		Reserved	
	0x110	All CPUs	AP17_START[31:12]; Reserved[11:1]; AP17_LOCK[0]			
	0x114	All CPUs	AP17_END[31:12]/Reserved[11:1]/AP17_COMMIT[0]			
	0x118	All CPUs	AP17_ACCESS			
	0x11C	All CPUs	AP17_CFG[15:0]		Reserved	
	0x120	All CPUs	AP18_START[31:12]; Reserved[11:1]; AP18_LOCK[0]			
	0x124	All CPUs	AP18_END[31:12]/Reserved[11:1]/AP18_COMMIT[0]			
	0x128	All CPUs	AP18_ACCESS			
	0x12C	All CPUs	AP18_CFG[15:0]		Reserved	
	0x130	All CPUs	AP19_START[31:12]; Reserved[11:1]; AP19_LOCK[0]			
	0x134	All CPUs	AP19_END[31:12]/Reserved[11:1]/AP19_COMMIT[0]			
	0x138	All CPUs	AP19_ACCESS			
	0x13C	All CPUs	AP19_CFG[15:0]		Reserved	
	0x140	All CPUs	AP20_START[31:12]; Reserved[11:1]; AP20_LOCK[0]			
	0x144	All CPUs	AP20_END[31:12]/Reserved[11:1]/AP20_COMMIT[0]			
	0x148	All CPUs	AP20_ACCESS			
	0x14C	All CPUs	AP20_CFG[15:0]		Reserved	
	0x150	All CPUs	AP21_START[31:12]; Reserved[11:1]; AP21_LOCK[0]			
	0x154	All CPUs	AP21_END[31:12]/Reserved[11:1]/AP21_COMMIT[0]			

Table 10-2. SECCFG Sector Memory Map (continued)

Section	Address Offset	CPUs	Bits 31:24	Bits 23:16	Bits 15:8	Bits 7:0
	0x158	All CPUs	AP21_ACCESS			
	0x15C	All CPUs	AP21_CFG[15:0]		Reserved	
	0x160	All CPUs	AP22_START[31:12]; Reserved[11:1]; AP22_LOCK[0]			
	0x164	All CPUs	AP22_END[31:12]/Reserved[11:1]/AP22_COMMIT[0]			
	0x168	All CPUs	AP22_ACCESS			
	0x16C	All CPUs	AP22_CFG[15:0]		Reserved	
	0x170	All CPUs	AP23_START[31:12]; Reserved[11:1]; AP23_LOCK[0]			
	0x174	All CPUs	AP23_END[31:12]/Reserved[11:1]/AP23_COMMIT[0]			
	0x178	All CPUs	AP23_ACCESS			
	0x17C	All CPUs	AP23_CFG[15:0]		Reserved	
	0x180	All CPUs	AP24_START[31:12]; Reserved[11:1]; AP24_LOCK[0]			
	0x184	All CPUs	AP24_END[31:12]/Reserved[11:1]/AP24_COMMIT[0]			
	0x188	All CPUs	AP24_ACCESS			
	0x18C	All CPUs	AP24_CFG[15:0]		Reserved	
	0x190	All CPUs	AP25_START[31:12]; Reserved[11:1]; AP25_LOCK[0]			
	0x194	All CPUs	AP25_END[31:12]/Reserved[11:1]/AP25_COMMIT[0]			
	0x198	All CPUs	AP25_ACCESS			
	0x19C	All CPUs	AP25_CFG[15:0]		Reserved	
	0x1A0	All CPUs	AP26_START[31:12]; Reserved[11:1]; AP26_LOCK[0]			
	0x1A4	All CPUs	AP26_END[31:12]/Reserved[11:1]/AP26_COMMIT[0]			
	0x1A8	All CPUs	AP26_ACCESS			
	0x1AC	All CPUs	AP26_CFG[15:0]		Reserved	
	0x1B0	All CPUs	AP27_START[31:12]; Reserved[11:1]; AP27_LOCK[0]			
	0x1B4	All CPUs	AP27_END[31:12]/Reserved[11:1]/AP27_COMMIT[0]			
	0x1B8	All CPUs	AP27_ACCESS			
	0x1BC	All CPUs	AP27_CFG[15:0]		Reserved	
	0x1C0	All CPUs	AP28_START[31:12]; Reserved[11:1]; AP28_LOCK[0]			
	0x1C4	All CPUs	AP28_END[31:12]/Reserved[11:1]/AP28_COMMIT[0]			
	0x1C8	All CPUs	AP28_ACCESS			
	0x1CC	All CPUs	AP28_CFG[15:0]		Reserved	
	0x1D0	All CPUs	AP29_START[31:12]; Reserved[11:1]; AP29_LOCK[0]			
	0x1D4	All CPUs	AP29_END[31:12]/Reserved[11:1]/AP29_COMMIT[0]			
	0x1D8	All CPUs	AP29_ACCESS			
	0x1DC	All CPUs	AP29_CFG[15:0]		Reserved	
	0x1E0	All CPUs	AP30_START[31:12]; Reserved[11:1]; AP30_LOCK[0]			
	0x1E4	All CPUs	AP30_END[31:12]/Reserved[11:1]/AP30_COMMIT[0]			
	0x1E8	All CPUs	AP30_ACCESS			
	0x1EC	All CPUs	AP30_CFG[15:0]		Reserved	
	0x1F0	All CPUs	AP31_START[31:12]; Reserved[11:1]; AP31_LOCK[0]			
	0x1F4	All CPUs	AP31_END[31:12]/Reserved[11:1]/AP31_COMMIT[0]			
	0x1F8	All CPUs	AP31_ACCESS			
	0x1FC	All CPUs	AP31_CFG[15:0]		Reserved	
	0x200	All CPUs	AP32_START[31:12]; Reserved[11:1]; AP32_LOCK[0]			
	0x204	All CPUs	AP32_END[31:12]/Reserved[11:1]/AP32_COMMIT[0]			
	0x208	All CPUs	AP32_ACCESS			

Table 10-2. SECCFG Sector Memory Map (continued)

Section	Address Offset	CPUs	Bits 31:24	Bits 23:16	Bits 15:8	Bits 7:0
	0x20C	All CPUs	AP32_CFG[15:0]		Reserved	
	0x210	All CPUs	AP33_START[31:12]; Reserved[11:1]; AP33_LOCK[0]			
	0x214	All CPUs	AP33_END[31:12]/Reserved[11:1]/AP33_COMMIT[0]			
	0x218	All CPUs	AP33_ACCESS			
	0x21C	All CPUs	AP33_CFG[15:0]		Reserved	
	0x220	All CPUs	AP34_START[31:12]; Reserved[11:1]; AP34_LOCK[0]			
	0x224	All CPUs	AP34_END[31:12]/Reserved[11:1]/AP34_COMMIT[0]			
	0x228	All CPUs	AP34_ACCESS			
	0x22C	All CPUs	AP34_CFG[15:0]		Reserved	
	0x230	All CPUs	AP35_START[31:12]; Reserved[11:1]; AP35_LOCK[0]			
	0x234	All CPUs	AP35_END[31:12]/Reserved[11:1]/AP35_COMMIT[0]			
	0x238	All CPUs	AP35_ACCESS			
	0x23C	All CPUs	AP35_CFG[15:0]		Reserved	
	0x240	All CPUs	AP36_START[31:12]; Reserved[11:1]; AP36_LOCK[0]			
	0x244	All CPUs	AP36_END[31:12]/Reserved[11:1]/AP36_COMMIT[0]			
	0x248	All CPUs	AP36_ACCESS			
	0x24C	All CPUs	AP36_CFG[15:0]		Reserved	
	0x250	All CPUs	AP37_START[31:12]; Reserved[11:1]; AP37_LOCK[0]			
	0x254	All CPUs	AP37_END[31:12]/Reserved[11:1]/AP37_COMMIT[0]			
	0x258	All CPUs	AP37_ACCESS			
	0x25C	All CPUs	AP37_CFG[15:0]		Reserved	
	0x260	All CPUs	AP38_START[31:12]; Reserved[11:1]; AP38_LOCK[0]			
	0x264	All CPUs	AP38_END[31:12]/Reserved[11:1]/AP38_COMMIT[0]			
	0x268	All CPUs	AP38_ACCESS			
	0x26C	All CPUs	AP38_CFG[15:0]		Reserved	
	0x270	All CPUs	AP39_START[31:12]; Reserved[11:1]; AP39_LOCK[0]			
	0x274	All CPUs	AP39_END[31:12]/Reserved[11:1]/AP39_COMMIT[0]			
	0x278	All CPUs	AP39_ACCESS			
	0x27C	All CPUs	AP39_CFG[15:0]		Reserved	
	0x280	All CPUs	AP40_START[31:12]; Reserved[11:1]; AP40_LOCK[0]			
	0x284	All CPUs	AP40_END[31:12]/Reserved[11:1]/AP40_COMMIT[0]			
	0x288	All CPUs	AP40_ACCESS			
	0x28C	All CPUs	AP40_CFG[15:0]		Reserved	
	0x290	All CPUs	AP41_START[31:12]; Reserved[11:1]; AP41_LOCK[0]			
	0x294	All CPUs	AP41_END[31:12]/Reserved[11:1]/AP41_COMMIT[0]			
	0x298	All CPUs	AP41_ACCESS			
	0x29C	All CPUs	AP41_CFG[15:0]		Reserved	
	0x2A0	All CPUs	AP42_START[31:12]; Reserved[11:1]; AP42_LOCK[0]			
	0x2A4	All CPUs	AP42_END[31:12]/Reserved[11:1]/AP42_COMMIT[0]			
	0x2A8	All CPUs	AP42_ACCESS			
	0x2AC	All CPUs	AP42_CFG[15:0]		Reserved	
	0x2B0	All CPUs	AP43_START[31:12]; Reserved[11:1]; AP43_LOCK[0]			
	0x2B4	All CPUs	AP43_END[31:12]/Reserved[11:1]/AP43_COMMIT[0]			
	0x2B8	All CPUs	AP43_ACCESS			
	0x2BC	All CPUs	AP43_CFG[15:0]		Reserved	

Table 10-2. SECCFG Sector Memory Map (continued)

Section	Address Offset	CPUs	Bits 31:24	Bits 23:16	Bits 15:8	Bits 7:0
	0x2C0	All CPUs	AP44_START[31:12]; Reserved[11:1]; AP44_LOCK[0]			
	0x2C4	All CPUs	AP44_END[31:12]/Reserved[11:1]/AP44_COMMIT[0]			
	0x2C8	All CPUs	AP44_ACCESS			
	0x2CC	All CPUs	AP44_CFG[15:0]		Reserved	
	0x2D0	All CPUs	AP45_START[31:12]; Reserved[11:1]; AP45_LOCK[0]			
	0x2D4	All CPUs	AP45_END[31:12]/Reserved[11:1]/AP45_COMMIT[0]			
	0x2D8	All CPUs	AP45_ACCESS			
	0x2DC	All CPUs	AP45_CFG[15:0]		Reserved	
	0x2E0	All CPUs	AP46_START[31:12]; Reserved[11:1]; AP46_LOCK[0]			
	0x2E4	All CPUs	AP46_END[31:12]/Reserved[11:1]/AP46_COMMIT[0]			
	0x2E8	All CPUs	AP46_ACCESS			
	0x2EC	All CPUs	AP46_CFG[15:0]		Reserved	
	0x2F0	All CPUs	AP47_START[31:12]; Reserved[11:1]; AP47_LOCK[0]			
	0x2F4	All CPUs	AP47_END[31:12]/Reserved[11:1]/AP47_COMMIT[0]			
	0x2F8	All CPUs	AP47_ACCESS			
	0x2FC	All CPUs	AP47_CFG[15:0]		Reserved	
	0x300	All CPUs	AP48_START[31:12]; Reserved[11:1]; AP48_LOCK[0]			
	0x304	All CPUs	AP48_END[31:12]/Reserved[11:1]/AP48_COMMIT[0]			
	0x308	All CPUs	AP48_ACCESS			
	0x30C	All CPUs	AP48_CFG[15:0]		Reserved	
	0x310	All CPUs	AP49_START[31:12]; Reserved[11:1]; AP49_LOCK[0]			
	0x314	All CPUs	AP49_END[31:12]/Reserved[11:1]/AP49_COMMIT[0]			
	0x318	All CPUs	AP49_ACCESS			
	0x31C	All CPUs	AP49_CFG[15:0]		Reserved	
	0x320	All CPUs	AP50_START[31:12]; Reserved[11:1]; AP50_LOCK[0]			
	0x324	All CPUs	AP50_END[31:12]/Reserved[11:1]/AP50_COMMIT[0]			
	0x328	All CPUs	AP50_ACCESS			
	0x32C	All CPUs	AP50_CFG[15:0]		Reserved	
	0x330	All CPUs	AP51_START[31:12]; Reserved[11:1]; AP51_LOCK[0]			
	0x334	All CPUs	AP51_END[31:12]/Reserved[11:1]/AP51_COMMIT[0]			
	0x338	All CPUs	AP51_ACCESS			
	0x33C	All CPUs	AP51_CFG[15:0]		Reserved	
	0x340	All CPUs	AP52_START[31:12]; Reserved[11:1]; AP52_LOCK[0]			
	0x344	All CPUs	AP52_END[31:12]/Reserved[11:1]/AP52_COMMIT[0]			
	0x348	All CPUs	AP52_ACCESS			
	0x34C	All CPUs	AP52_CFG[15:0]		Reserved	
	0x350	All CPUs	AP53_START[31:12]; Reserved[11:1]; AP53_LOCK[0]			
	0x354	All CPUs	AP53_END[31:12]/Reserved[11:1]/AP53_COMMIT[0]			
	0x358	All CPUs	AP53_ACCESS			
	0x35C	All CPUs	AP53_CFG[15:0]		Reserved	
	0x360	All CPUs	AP54_START[31:12]; Reserved[11:1]; AP54_LOCK[0]			
	0x364	All CPUs	AP54_END[31:12]/Reserved[11:1]/AP54_COMMIT[0]			
	0x368	All CPUs	AP54_ACCESS			
	0x36C	All CPUs	AP54_CFG[15:0]		Reserved	
	0x370	All CPUs	AP55_START[31:12]; Reserved[11:1]; AP55_LOCK[0]			

Table 10-2. SECCFG Sector Memory Map (continued)

Section	Address Offset	CPUs	Bits 31:24	Bits 23:16	Bits 15:8	Bits 7:0
	0x374	All CPUs	AP55_END[31:12]/Reserved[11:1]/AP55_COMMIT[0]			
	0x378	All CPUs	AP55_ACCESS			
	0x37C	All CPUs	AP55_CFG[15:0]		Reserved	
	0x380	All CPUs	AP56_START[31:12]; Reserved[11:1]; AP56_LOCK[0]			
	0x384	All CPUs	AP56_END[31:12]/Reserved[11:1]/AP56_COMMIT[0]			
	0x388	All CPUs	AP56_ACCESS			
	0x38C	All CPUs	AP56_CFG[15:0]		Reserved	
	0x390	All CPUs	AP57_START[31:12]; Reserved[11:1]; AP57_LOCK[0]			
	0x394	All CPUs	AP57_END[31:12]/Reserved[11:1]/AP57_COMMIT[0]			
	0x398	All CPUs	AP57_ACCESS			
	0x39C	All CPUs	AP57_CFG[15:0]		Reserved	
	0x3A0	All CPUs	AP58_START[31:12]; Reserved[11:1]; AP58_LOCK[0]			
	0x3A4	All CPUs	AP58_END[31:12]/Reserved[11:1]/AP58_COMMIT[0]			
	0x3A8	All CPUs	AP58_ACCESS			
	0x3AC	All CPUs	AP58_CFG[15:0]		Reserved	
	0x3B0	All CPUs	AP59_START[31:12]; Reserved[11:1]; AP59_LOCK[0]			
	0x3B4	All CPUs	AP59_END[31:12]/Reserved[11:1]/AP59_COMMIT[0]			
	0x3B8	All CPUs	AP59_ACCESS			
	0x3BC	All CPUs	AP59_CFG[15:0]		Reserved	
	0x3C0	All CPUs	AP60_START[31:12]; Reserved[11:1]; AP60_LOCK[0]			
	0x3C4	All CPUs	AP60_END[31:12]/Reserved[11:1]/AP60_COMMIT[0]			
	0x3C8	All CPUs	AP60_ACCESS			
	0x3CC	All CPUs	AP60_CFG[15:0]		Reserved	
	0x3D0	All CPUs	AP61_START[31:12]; Reserved[11:1]; AP61_LOCK[0]			
	0x3D4	All CPUs	AP61_END[31:12]/Reserved[11:1]/AP61_COMMIT[0]			
	0x3D8	All CPUs	AP61_ACCESS			
	0x3DC	All CPUs	AP61_CFG[15:0]		Reserved	
	0x3E0	All CPUs	AP62_START[31:12]; Reserved[11:1]; AP62_LOCK[0]			
	0x3E4	All CPUs	AP62_END[31:12]/Reserved[11:1]/AP62_COMMIT[0]			
	0x3E8	All CPUs	AP62_ACCESS			
	0x3EC	All CPUs	AP62_CFG[15:0]		Reserved	
	0x3F0	All CPUs	AP63_START[31:12]; Reserved[11:1]; AP63_LOCK[0]			
	0x3F4	All CPUs	AP63_END[31:12]/Reserved[11:1]/AP63_COMMIT[0]			
	0x3F8	All CPUs	AP63_ACCESS			
	0x3FC	All CPUs	AP63_CFG[15:0]		Reserved	

Table 10-2. SECCFG Sector Memory Map (continued)

Section	Address Offset	CPUs	Bits 31:24	Bits 23:16	Bits 15:8	Bits 7:0
Flash Write/ Erase Protection	0x400 to 0x667	N/A	Reserved			
	0x668	CPU1 only	Reserved	WEPROT_CODE _COMMIT	WEPROT_CODE _LOCK	WEPROT_CODE _BANKS
	0x66C	CPU1 only	Reserved	WEPROT_DATA _COMMIT	WEPROT_DATA _LOCK	WEPROT_DATA _BANKS
	0x670	CPU1 only	WEPROT_FLC1_B0_A			
	0x674	CPU1 only	WEPROT_FLC1_ B0_COMMIT	WEPROT_FLC1_ B0_LOCK	Reserved	
	0x678	CPU1 only	WEPROT_FLC1_B0_B			
	0x67C	N/A	Reserved			
	0x680	CPU1 only	WEPROT_FLC1_B2_A			
	0x684	CPU1 only	WEPROT_FLC1_ B2_COMMIT	WEPROT_FLC1_ B2_LOCK	Reserved	
	0x688	CPU1 only	WEPROT_FLC1_B2_B			
	0x68C	N/A	Reserved			
	0x690	CPU1 only	WEPROT_FLC2_B0_A			
	0x694	CPU1 only	WEPROT_FLC2_ B0_COMMIT	WEPROT_FLC2_ B0_LOCK	Reserved	
	0x698	CPU1 only	WEPROT_FLC2_B0_B			
	0x69C	N/A	Reserved			
	0x6A0	CPU1 only	WEPROT_FLC2_B2_A			
	0x6A4	CPU1 only	WEPROT_FLC2_ B2_COMMIT	WEPROT_FLC2_ B2_LOCK	Reserved	
	0x6A8	CPU1 only	WEPROT_FLC2_B2_B			
	0x6AC to 0x6B7	N/A	Reserved			
Flash Update Configuration	0x6B8	CPU1 only	ZONE2_CFG		ZONE1_CFG	
	0x6BC	CPU1 only	SECCFG_UPDATE_CFG		ZONE3_CFG	

Table 10-2. SECCFG Sector Memory Map (continued)

Section	Address Offset	CPUs	Bits 31:24	Bits 23:16	Bits 15:8	Bits 7:0
Debug Passwords	0x6C0	CPU1 only	ZONE1_PDBGPSWD[31:0]			
	0x6C4	CPU1 only	ZONE1_PDBGPSWD[63:32]			
	0x6C8	CPU1 only	ZONE1_PDBGPSWD[95:64]			
	0x6CC	CPU1 only	ZONE1_PDBGPSWD[127:96]			
	0x6D0 to 0x6DF	N/A	Reserved			
	0x6E0	CPU1 only	ZONE1_FDBGPSWD[31:0]			
	0x6E4	CPU1 only	ZONE1_FDBGPSWD[63:32]			
	0x6E8	CPU1 only	ZONE1_FDBGPSWD[95:64]			
	0x6EC	CPU1 only	ZONE1_FDBGPSWD[127:96]			
	0x6F0 to 0x6FF	N/A	Reserved			
	0x700	CPU1 only	ZONE2_PDBGPSWD[31:0]			
	0x704	CPU1 only	ZONE2_PDBGPSWD[63:32]			
	0x708	CPU1 only	ZONE2_PDBGPSWD[95:64]			
	0x70C	CPU1 only	ZONE2_PDBGPSWD[127:96]			
	0x710 to 0x71F	N/A	Reserved			
	0x720	CPU1 only	ZONE2_FDBGPSWD[31:0]			
	0x724	CPU1 only	ZONE2_FDBGPSWD[63:32]			
	0x728	CPU1 only	ZONE2_FDBGPSWD[95:64]			
	0x72C	CPU1 only	ZONE2_FDBGPSWD[127:96]			
	0x730 to 0x73F	N/A	Reserved			
	0x740	CPU1 only	ZONE3_PDBGPSWD[31:0]			
	0x744	CPU1 only	ZONE3_PDBGPSWD[63:32]			
	0x748	CPU1 only	ZONE3_PDBGPSWD[95:64]			
	0x74C	CPU1 only	ZONE3_PDBGPSWD[127:96]			
	0x750 to 0x75F	N/A	Reserved			
	0x760	CPU1 only	ZONE3_FDBGPSWD[31:0]			
	0x764	CPU1 only	ZONE3_FDBGPSWD[63:32]			
	0x768	CPU1 only	ZONE3_FDBGPSWD[95:64]			
	0x76C	CPU1 only	ZONE3_FDBGPSWD[127:96]			
	0x770 to 0x77F	N/A	Reserved			
	0x780	CPU1 only	C29DBGEN_DBGPSWD[31:0]			
	0x784	CPU1 only	C29DBGEN_DBGPSWD[63:32]			
	0x788	CPU1 only	C29DBGEN_DBGPSWD[95:64]			
	0x78C	CPU1 only	C29DBGEN_DBGPSWD[127:96]			
	0x790 to 0x79F	N/A	Reserved			
Debug Configuration	0x7A0	CPU1 only	DEBUG_CFG_ COMMIT	DEBUG_CFG_ LOCK	DEBUG_CFG	
	0x7A4	CPU1 only	Reserved		ENABLE_XTAL	Reserved
Boot Configuration	0x7A8	CPU1 only	BOOTPIN_CFG			
	0x7AC	CPU1 only	Reserved	BOOT_LOADER_ LOCK	ERROR_STS_PIN	POST_EN
	0x7B0	CPU1 only	RAMOPEN_ COMMIT	RAMOPEN_ LOCK	EMU_BOOTEN	Reserved
	0x7B4	N/A	Reserved			

Table 10-2. SECCFG Sector Memory Map (continued)

Section	Address Offset	CPUs	Bits 31:24	Bits 23:16	Bits 15:8	Bits 7:0
SECCFG Management	0x7B8	CPU1 only	SECCFG_STATUS[31:0]			
	0x7BC	CPU1 only	SECCFG_STATUS[63:32]			
	0x7C0	CPU1 only	SECCFG_UPDATE_CTR[31:0]			
	0x7C4	CPU1 only	SECCFG_UPDATE_CTR[63:32]			
CPU Boot Configuration	0x7C8	CPU1 only	BOOTDEF_LOW			
	0x7CC	CPU1 only	BOOTDEF_HIGH			
CPU LINK/ STACK Configuration	0x7D0	All CPUs	STACK3_CFG	Reserved		
	0x7D4	All CPUs	STACK7_CFG	STACK6_CFG	STACK5_CFG	STACK4_CFG
	0x7D8	All CPUs	LINK3_CFG	Reserved		
	0x7DC	All CPUs	LINK7_CFG	LINK6_CFG	LINK5_CFG	LINK4_CFG
	0x7E0	All CPUs	LINK11_CFG	LINK10_CFG	LINK9_CFG	LINK8_CFG
	0x7E4	All CPUs	LINK15_CFG	LINK14_CFG	LINK13_CFG	LINK12_CFG
	0x7E8 to 0x7EF	N/A	Reserved			
SECCFG Format/CRC	0x7F0	CPU1 only	Reserved			SECCFG_FORMAT_ VER
	0x7F4	CPU1 only	SECCFG_CRC			
Critical Configuration	0x7F8	CPU1 only	SSU_MODE			
	0x7FC	CPU1 only	Reserved		UPP_REVISION	

10.8.5 SECCFG CRC

As an added security measure, the device boot ROM verifies the contents of the active CPU1 SECCFG sector using a CRC32 algorithm before loading the settings stored in the SECCFG sector into SSU registers. If the CRC check fails, the boot process is aborted. The SECCFG CRC is computed using the following parameters:

- **CRC Polynomial:** 0x04C1 1DB7 (CRC-32/ISO-HDLC)
- **Initial Value:** 0xFFFF FFFF
- **Final XOR Value:** 0xFFFF FFFF
- **Address Ranges:**
  1. 0x000 to 0x6BF
  2. 0x7A0 to 0x7F3
  3. 0x7F8 to 0x7FF

10.9 Flash Write/Erase Access Control

In addition to LINK, STACK and ZONE IDs, the SSU includes a Flash semaphore (FLSEM) to grant exclusive access to the Flash controller for programming, program verify, erase, and erase verify operations by a given initiator at any point in time. The Flash semaphore locks all data accesses to the Flash controller registers to a specific CPU and LINK. Upon receiving a command, the Flash Controller sends a permission request out through the Flash Crossbar port (FXB) to the SSU. The SSU then determines, based on the current security context and the FLSEM owner, whether to grant permission to execute the Flash command. Flash Controller operations are not permitted when the FLSEM has no owner.

The LINK, ZONE and CPU fields in the FLSEMSTAT register specify the current owner of the Flash semaphore. In addition, code running from any LINK can confirm ownership of the semaphore by reading the FLSEMSTAT.MATCH bit. This bit returns 1 only if the LINKID and CPUID of the code performing the read match the current owner of the Flash semaphore. The FLSEMSTAT.ASSIGNED bit indicates whether the Flash semaphore is currently owned.

To grab the Flash semaphore:
1. Write 1 to the FLSEMREQ register.
2. Confirm ownership of the Flash semaphore by checking that FLSEMSTAT.MATCH = 1.

To release the Flash semaphore:
1. Write 1 to the FLSEMCLR register.
2. Confirm the Flash semaphore has been released by checking that FLSEMSTAT.MATCH = 0.

Note

With the Flash semaphore mechanism, configuring an access protection range to cover the Flash Controller registers is not necessary. Once the Flash semaphore has been assigned to a specific CPU ID and LINK ID, all accesses to the Flash Controller registers are restricted to that LINK, until the semaphore owner LINK releases the FLSEM by writing 1 to the FLSEMCLR register. In this way, the designated LINK can lock the Flash controller for exclusive self use, enabling code security and preventing access by other LINKs.

10.9.1 Permanent Flash Lock (Write/Erase Protection)

The SSU provides a mechanism to globally block program and erase operations to any Flash bank. To block all program and erase operations on code Flash banks throughout the device, write 1 to the WEPROT_CODE_BANKS.PROT register. To block program and erase operations to data Flash banks, write 1 to the WEPROT_DATA_BANKS.PROT register.

These protections can be hardened for safety or security purposes with the SSU's LOCK/COMMIT scheme using the following registers: WEPROT_CODE_BANKS_LOCK, WEPROT_CODE_BANKS_COMMIT, WEPROT_DATA_BANKS_LOCK, and WEPROT_DATA_BANKS_COMMIT.

Additionally, individual Flash sectors can be protected from write and erase operations using the WEPROT_FLC*x*_B*y*_* registers. For each bank, there are three registers. The first, WEPROT_FLC*x*_B*y*_A, contains individual protection bits for the first 32 sectors in bank *y* of Flash Controller *x*. The other two, WEPROT_FLC*x*_B*y*_B and WEPROT_FLC*x*_B*y*_C, define protections for the remaining Flash sectors in the bank, in groups of 8 contiguous sectors. These registers also have LOCK and COMMIT mechanisms, similar to WEPROT_CODE_BANKS and WEPROT_DATA_BANKS.

All the Flash protection registers are mapped to the SECCFG sector, so that Flash protections defined in SECCFG are automatically loaded by the boot ROM at device start-up. Using this mechanism, Flash bank or sector protections can be locked and committed before the application starts, and critical Flash contents remain unmodifiable throughout the current power-on cycle.

10.9.2 Updating Flash MAIN Sectors

For each ZONE, the user can configure a firmware update owner CPU and LINK using the FWU_CPU and FWU_LINK fields in the ZONEx_CFG register. The following restrictions apply when updating code in the Flash MAIN sectors:

- When in SSUMODE3, the firmware update owner CPU and LINK defined in the ZONEx_CFG register is the only LINK allowed to program and erase the Flash MAIN sectors. In addition, ZONEx_CFG.UPDATE_EN must be set to 0xC to enable updates.
- In SSUMODE1 and SSUMODE2, Flash write/erase protections are inactive as all ZONES are enabled for full debug. In these modes, Flash memory can always be programmed or erased. This enables easy code development.

10.9.3 Firmware-Over-The-Air Updates (FOTA)

This device supports Firmware-Over-The-Air (FOTA) updates using a Flash bank-swapping mechanism. The FOTA process is managed and controlled using the Flash Controller BANKMAP register and the BANKMGMT sector in each bank. Figure 10-5 shows an example of how banks are swapped.

When bank swap is active, all sectors of a Flash bank are swapped, including the SECCFG and BANKMGMT sectors.

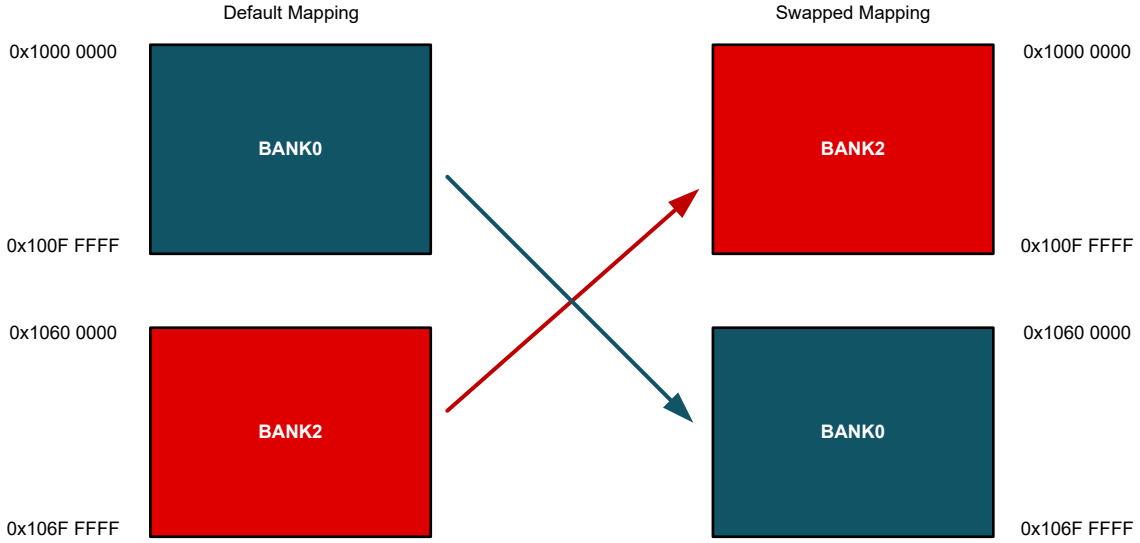


Figure 10-5. Bank Swap Mapping Example

The FOTA concept is designed to enable programming of inactive Flash banks while the application continues to run from the active banks without interruption. The update process is comprised of the following steps:

1. **Image Creation and Deployment:** The user creates an application update image. This image can be transferred into the device using a Flash kernel, over a communications interface such as CAN or UART. The update image is typically delivered in chunks, since the RAM available is not large enough to hold an entire Flash image. For security, the update image can be authenticated using cryptographic methods such as a digital signature algorithm.
2. **Installation Phase:** The Flash kernel downloads Flash chunks, authenticates them, and programs them into the Flash update region (FRI-3). This phase consists of three steps:
   a. Erase the inactive banks by addressing the update region with a Flash erase command.
   b. Download, authenticate/decrypt and program updated firmware into the Flash update region.
   c. Issue Flash API command to update BANKMGMT sector of inactive banks to be active and newer than the currently active banks. The Flash API commands are executed in the following order:
      i. (CPU1 only) Program inactive bank's BANKMODE to the expected/new BANKMODE value. Note that changing the BANKMODE during a firmware update is possible.
      ii. Program the inactive bank's BANK_UPDATE_CTR to be one less than the active bank's BANK_UPDATE_CTR.
      iii. Program the inactive bank's BANK_STATUS to 0x55555555_55555555.
3. **Activation Phase:** The update manager waits for the application to reach a safe state (or brings the application to a safe state), then issues a full device reset (XRSn). A reset can be issued by software running in LINK2 by writing to the DEV_CFG_REGS.SIMRESET register. During device startup, the boot ROM reevaluates the BANKMGMT sectors, and configures BANKMAP to reflect the new update. After a timeout period, the FOTA manager requests the active firmware revision from the device. If the expected new version is not returned, the FOTA manager re-issues the update command, since the update has not completed successfully.

4. (Optional) **Rollback:** If desired, the user can return to the most recent (currently inactive) application firmware version. The rollback service erases and reprograms the current inactive BANKMGMT sector to be active and have a newer update counter value than the current active sector. This allows a rollback to the previous firmware version without having to re-download from an external interface. The rollback steps are executed in the following order:
   a. Authenticate the code in the inactive bank to confirm the code's validity and integrity.
   b. Erase the inactive bank's BANKMGMT sector.
   c. (CPU1 only) Program inactive bank's BANKMODE to the correct BANKMODE value.
   d. Program the inactive bank's BANK_UPDATE_CTR to one less than the active bank's BANK_UPDATE_CTR.
   e. Program the inactive bank's BANK_STATUS to 0x55555555_55555555.
   f. After bringing the application to a safe state, issue a device XRSn reset to activate the old firmware image.
5. (Optional) **Anti-Rollback:** To prevent any possibility of a rollback to the previous firmware version, invalidate the old firmware image completely by erasing the inactive bank's MAIN sectors.

10.9.4 Updating Flash SECCFG Sectors

The following security restrictions apply to updates of the SECCFG sectors:

- In SSUMODE1, all code running from any LINK has permission to update the SECCFG sectors, since ZONE1 is enabled for full debug. This enables ease of code development.
- In SSUMODE2, all code running from any LINK has permission to update SECCFG sectors, since ZONE1 is enabled for full debug.
- In SSUMODE3, only the SECCFG update owner defined in SECCFG_UPDATE_CFG has permission to update SECCFG sectors. In addition, SECCFG_UPDATE_CFG.UPDATE_EN must be set to 0xC to enable updates to SECCFG when operating in SSUMODE3.

Each CPU has two SECCFG sectors for storing User Protection Policy (UPP). One sector holds the active policy, while the other is available for updating. The SECVALID register specifies which of the two SECCFG sectors is active for each CPU subsystem.

When updating the SSU configuration, always program the SECCFG image to the reserve (B2/B3) SECCFG sector address, regardless of which sector is currently active. The SSU automatically routes the reserve address to the current inactive sector when a Flash program or erase command is performed. The SECCFG image base addresses are defined in Table 10-3. However, CPU read operations must be addressed to the correct Flash read port address, as defined in Section 10.8.3.

Table 10-3. CPU1 SECCFG Update Address Table

	CPU1/CPU2	CPU3/CPU4	CPU1/CPU2 (FOTA/LFU)	CPU3/CPU4 (FOTA/LFU)
Erase/Program	0x10D8 5000	0x10D9 5000	0x10D9 9000	0x10D9 D000
Read	Refer to Section 10.8.3			

10.9.5 Reading Flash SECCFG Sectors

As a countermeasure against overclocking attacks, all reads to Flash SECCFG sectors are hard-coded to 10 cycles (9 wait states), regardless of the current Flash wait state configuration.

10.10 RAMOPEN Feature

The RAMOPEN mode provides a means for bootloader applications to exclusively access RAM for the purpose of storing code or data downloaded from an external source. RAMOPEN can only be initiated by two sources:

1. Direct JTAG writes to the RAMOPENFRC register. This gives the debugger the ability to open RAM access for writing before downloading code to the RAM. Debugger writes to LINK1 RAM are only allowed in SSUMODE1, in SSUMODE2, or when ZONE1 is enabled for full debug.
2. Boot loader code running in LINK1.

RAMOPEN cannot be initiated by code running from any of the user LINKs (2 and above).

When RAMOPEN is opened, the following sequence occurs:

1. The hardware erases the RAMs using RAMINIT.
2. Any write to RAM that was initiated prior to or during RAMINIT is aborted. Write accesses can only update RAM after RAMINIT completes and RAMOPEN status becomes active.
3. LINK1 gains read, write and execute permissions for RAMOPENed memories. All other LINKs have no access to any of the RAMs.
4. The SSU sets RAMOPENSTAT.LINK1_RAMOPENS to 1 to indicate RAMOPEN is now active.

After the intended use for RAMOPEN completes, return the RAM back to the user application by writing 1 to the RAMOPENCLR.CLEAR_LINK1 register. Once the bit has been written, if the RAMOPENFRC.WIPE_ON_LINK1_CLR bit was set, the device erases the RAM content for all RAMOPENed memories using the RAMINIT feature, and then clears the RAMOPENSTAT.LINK1_RAMOPENS bit. If the RAMOPENFRC.WIPE_ON_LINK1_CLR bit was not set, then the RAM contents are preserved when RAM is returned to the user application. This feature gives user bootloaders (for example, firmware updaters) the ability to automatically wipe sensitive data or code from the RAM after the bootloader operation completes.

The following sequence occurs when RAMOPENCLR.CLEAR_LINK1 is written:

1. If RAMOPENFRC.WIPE_ON_LINK1_CLR is set, then the RAMs are erased using RAMINIT.
2. Any write to RAM that was initiated prior to or during RAMINIT is aborted. Only writes initiated after RAMINIT completes and RAMOPEN status is set to inactive update the RAM.
3. RAM access permissions are allocated according to the configured user protection policy (UPP).
4. The SSU sets RAMOPENSTAT.LINK1_RAMOPENS to 0 to indicate RAMOPEN is now inactive.

Note

In RAMOPEN mode, unauthorized code can be loaded into memory as LINK1 code. Any ZONEs that permit firmware updates from the LINK1 can be updated without user code control. The most secure way to perform a firmware update is to lock boot loaders down and use a secure firmware update algorithm.

10.11 Debug Authorization

There are two levels of debug authorization provided by the SSU:

- Global JTAG authorization (C29DBGEN)
- ZONE debug authorization

Debug access control features are active in SSUMODE3. In SSUMODE1 and SSUMODE2, JTAG is automatically enabled, and all user ZONEs are enabled for full debug.

10.11.1 Global CPU Debug Enable

The global debug enable control gates debug access to all C29x CPU test access ports (TAPs). When C29DBGEN is inactive (set to 0), none of the CPU cores are accessible directly using JTAG; only communication with the device Security Access Port (SEC-AP) handler is permitted.

When present, the HSM can also control debug access to the C29x CPUs by writing to the C29DBGEN or ZONE_DBGEN registers in the SSU.

Note

C29DBGEN only controls debug access to the C29x CPUs, and has no effect on HSM debug access. C29DBGEN is different from the HSM global JTAG disable control (JTAG_DIS) which blocks access to all device debug ports, including both HSM and C29x CPU subsystems.

10.11.2 ZONE Debug

In addition to the global C29x debug enable control, each ZONE has a separate enable setting, for more granular debug access control. ZONEs can span one or more CPUs, and are orthogonal to individual CPU TAP enables. ZONEs can be used to separate debug authorization across different parts of the device between different users, with different authentication credentials established for each user. When a user performs a debug access to a memory region, the SSU checks to see if at least one LINK with the requested access to the region belongs a ZONE that has been enabled for full debug. If no LINKs match this condition, then the SSU blocks the debug access, and notifies the C29x SEC-AP handler that a debug read or write error occurred, depending on the type of access attempted.

10.11.3 Authentication for Debug Access

The DEBUG_CFG register controls the authentication method used to control debug access at the C29DBGEN level and each of the user ZONE debug levels. For each debug access control, there are three valid configuration options:

- Password-based authentication
- CPU-based authentication
- Debug disabled

The DEBUG_CFG register has LOCK/COMMIT protections to prevent unintended or unauthorized changes to the debug authentication scheme. To lock or commit these settings, write to the DEBUG_CFG_LOCK and DEBUG_CFG_COMMIT registers as required.

The current debug status for C29DBGEN and each of the user debug ZONEs can be determined by connecting to the SEC-AP and reading the DEBUG_ENABLE_STATUS register. For more details, see the Debug Controller chapter.

Devices shipped from the TI factory have password-based authentication active at boot-up by default. To change the startup debug authentication mode, update the DEBUG_CFG field in the SECCFG sector accordingly.

10.11.3.1 Password-based Authentication

When password-based authorization is enabled for C29DBGEN or a user ZONE, the associated register bit field (C29DBGEN.ENA, ZONE_DBG.ZONEx) is ignored by the SSU. Instead, the user scans a 128-bit password into the appropriate internal key register using the SEC-AP, and the SSU hardware compares the scanned-in password to a predefined password that is preloaded from the SECCFG sector during device boot. If the password matches, then the SSU opens debug access to the requested resource.

There are two levels of ZONE debug authorization that can be enabled by password authentication: full debug, and partial debug. These are controlled using the full debug password and partial debug password respectively. When a user scans in a password for debug authorization, the SSU performs the following sequence:

1. Compare scanned-in password to full debug password. If matching, then open the ZONE for full debug.
2. If the previous compare resulted in a mismatch, then compare the scanned-in password to the partial debug password. If matching, then open the ZONE for partial debug.
3. If the previous compare resulted in a mismatch, then the ZONE remains closed.

To enable C29DBGEN or open debug access to a zone when using password-based authentication, scan the appropriate command as defined in Table 10-4 into the SEC-AP PASSWORD_ID register, and then scan in the 128-bit password into the SEC-AP password registers (PASSWORD1, PASSWORD2, PASSWORD3, PASSWORD4). Note that there is a single password scan required for both full and partial ZONE debug. The SSU first compares the scanned in password to the full debug password. If there is a match, then full debug is enabled for the ZONE. If there is a mismatch, then the SSU compares the scanned in password to the partial debug password to determine whether to enable partial debug for the ZONE.

Table 10-4. Password Scan Commands

Value	Command
0x0	Reserved for internal use
0x1	ZONE1 Debug Password Scan
0x2	ZONE2 Debug Password Scan
0x3	ZONE3 Debug Password Scan
0x4	C29DBGEN Unlock Password Scan
Others	Reserved for future use

For C29DBGEN, there is only one password that controls debug access. When the user scans in a password matching the C29DBGEN password, the SSU unlocks the Test Access Ports (TAPs) for all application CPUs in the device. The HSM also has control over the application CPU TAPs. When the HSM is present, both C29DBGEN and the corresponding HSM debug enable control must be active for debug access to the C29x CPU TAP to be opened. If either C29DBGEN or the HSM denies access, then the C29x CPU TAP remains closed.

Note

The SSU has no control over HSM resources. C29DBGEN only controls debug access to C29x CPUs. For information on how to enable debug access to the HSM, see the *F29x Security Hardware User Guide*.

C29DBGEN does not override ZONE debug authorization. Even with C29DBGEN active, a password-protection ZONE must still be unlocked for the debugger to access any memory-mapped resources in that ZONE. C29DBGEN only controls access to the CPU TAPs.

10.11.3.2 CPU-based Authentication

CPU-based debug authentication enables users to write customized code for debug authorization, instead of a simple password scan and compare. In this mode, the application can execute challenge-response authentication schemes using cryptographic algorithms, including schemes involving public/private key encryption pairs. Challenge-response authentication provides stronger security by eliminating the need to store a common password across all devices which, if stolen, potentially compromises security for existing deployed devices.

When CPU-based authentication is selected in DEBUG_CFG for a particular ZONE or for C29DBGEN, the corresponding register bit fields (ZONE_DBGEN.ZONEx, C29DBGEN.ENA) become valid, and the password comparison is ignored. The ZONE_DBGEN and C29DBGEN registers are only accessible by CPU1.LINK2 and the HSM. The application authorization code, running on CPU1.LINK2 or on the HSM, can read values scanned into the SEC-AP through the debugger, and through the interactive challenge-response process determine if and when to enable debug by writing to the C29DBGEN or ZONE_DBGEN registers.

10.12 Hardcoded Protections

The user has the ability to configure all LINKs, STACKs and ZONEs that are owned by the user. However, certain system resources have protections that are fixed in hardware and cannot be altered:

- LINK0 belongs to STACK0, and STACK0 belongs to ZONE0. These are TI internal resources and cannot be accessed by user code. User code cannot assign any user LINKs to STACK0, or user STACKs to ZONE0.
- LINK1 for each CPU always belongs to STACK1.
- STACK1 for each CPU always belongs to ZONE1.
- LINK2 for each CPU always belongs to STACK2.
- STACK2 for each CPU always belongs to ZONE1.
- In SSUMODE3, the Flash SECCFG sector can only be read by:
  - The SECCFG update owner defined in SECCFG_UPDATE_CFG, or
  - The HSM, if present and enabled.
- Code instruction fetches to M0 memory always generate a fault.
- Data reads of M0 RAM are allowed for all CPUs and all LINKs.
- Data writes to M0 RAM are only allowed for CPU1.LINK2.
- Certain memory and peripheral regions are restricted to one or more of the special-purpose LINKs (LINK0, LINK1 and LINK2). For details on these non-APR-based memory protections, refer to the memory-map in the device data sheet.

10.13 SSU Register Access Permissions

SSU register permissions vary depending on category and function. CPU1 is the primary owner of all SSU registers, including those for all other CPUs. Most registers are automatically loaded at device startup; however, some are designated for run-time operation, for example, FLSEM. The tables in this section describe the access permissions for each register, organized by register groups. For register access permission tables that do not include a column for the HSM, there is no HSM access permitted to these registers.

Accesses to SSU registers that are denied generate a fault to the ESM.

Note

When lockstep mode is enabled, the register access permissions for the secondary CPU are always identical to the access permissions for the primary CPU. For instance, if CPU2 is present and in lockstep with CPU1, then CPU2's access permissions are the same as CPU1's. The CPU2 permission tables do not apply. For more information on SSU operation in lockstep mode, see Section 10.6.1.

In SSUMODE1, CPU register read access is granted if any LINK is enabled for hardcoded read permissions. Similarly, in SSUMODE1 CPU register write access is granted if any LINK is enabled for hardcoded write permissions. This rule does not supersede Flash semaphore ownership rules.

10.13.1 Permissions for SSU General Control Registers

Table 10-5 describes access permissions for the SSU_GEN_REGS group of registers.

Table 10-5. Access Permissions for SSU General Control Registers

Register	Register Load Method	Boot	CPU1				CPU2+			HSM
			CPU1 LINK1	CPU1 LINK2	CPU1 LINK3+	CPU1 Debug	CPU2+ LINK0,1, and 3+	CPU2+ LINK2	CPU2+ Debug	
REVISION	None	R-only	R-only	R-only	R-only	R-only(1)	R-only	R-only	R-only(1)	No
UPP_ REVISION	Boot	R/W	R-only	R-only	R-only	R-only(1)	No	No	No	No
SSUMODE	Boot	R/W	R-only	R-only	R-only	R-only(1)	R-only	R-only	R-only(1)	No
LINK2_AP_ OVERRIDE	None	R-only	R-only	R/W	R-only	R/W(2)	R-only	R/W	R/W(2)	No
BOOTMODE_ STAT	Boot	R/W	R-only	R-only	R-only	R-only	R-only	R-only	R-only	No
EMU_BOOTPIN_ CONFIG	None	R-only	R-only	R-only	R-only	R/W(2)	R-only	R-only	R/W(2)	No
EMU_BOOT_ DIAG	None	R-only	R-only	R-only	R-only	R/W(2)	R-only	R-only	R/W(2)	No
EMU_BOOT_ CLKCFG	None	R-only	R-only	R-only	R-only	R/W(2)	R-only	R-only	R/W(2)	No
EMU_BOOTEN	Boot	R/W	R-only	R-only	R-only	R-only	R-only	R-only	R-only	No
RAMOPEN_ LOCK	Boot	R/W	R-only	R-only	R-only	R-only(1)	R-only	R-only	R-only(1)	No
RAMOPEN_ COMMIT	Boot	R/W	R-only	R-only	R-only	R-only(1)	R-only	R-only	R-only(1)	No
CPUID	None	R-only	R-only	R-only	R-only	R-only	R-only	R-only	R-only	No
BANKMAP	Boot	R/W	R-only	R-only(4)	R-only	R/W(2)	R-only	R-only	R/W(2)	R/W
BANKMAP_ LOCK	Boot	R/W	R-only	R-only(4)	R-only	R/W(2)	R-only	R-only	R/W(2)	R/W
BANKMAP_ COMMIT	None	R/W	R-only	R-only(4)	R-only	R/W(2)	R-only	R-only	R/W(2)	R/W
BANKMODE	Boot	R/W	R-only	R-only	R-only	R/W(2)	R-only	R-only	R/W(2)	R/W

Table 10-5. Access Permissions for SSU General Control Registers (continued)

Register	Register Load Method	Boot	CPU1				CPU2+			HSM
			CPU1 LINK1	CPU1 LINK2	CPU1 LINK3+	CPU1 Debug	CPU2+ LINK0,1, and 3+	CPU2+ LINK2	CPU2+ Debug	
BANKMODE_ LOCK	Boot	R/W	R-only	R-only	R-only	R/W(2)	R-only	R-only	R/W(2)	R/W
BANKMODE_ COMMIT	None	R/W	R-only	R-only	R-only	R/W(2)	R-only	R-only	R/W(2)	R/W
SECCFG_ UPDATE_CFG	Boot	R/W	R-only	R-only	R-only	R-only(1)	R-only	R-only	R-only(1)	No
PROG_ BANKMODE	None	R/W	R/W	R/W	R/W	No	R/W	R/W	No	R/W
SECVALID	HSM	R/W	R-only	R-only	R-only	R-only(1)	R-only	R-only	R-only(1)	R/W
SECVALID_ LOCK	HSM	R/W	R-only	R-only	R-only	R-only(1)	R-only	R-only	R-only(1)	R/W
SECVALID_ COMMIT	None	R/W	R-only	R-only	R-only	R-only(1)	R-only	R-only	R-only(1)	R/W
ZONEx_CFG	Boot	R/W	R-only	R-only	R-only	R-only(1)	R-only	R-only	R-only(1)	No
DEBUG_CFG	Boot	R/W	R-only	R/W	R-only	R-only	R-only	R-only	R-only	R/W
DEBUG_CFG_ LOCK	Boot	R/W	R-only	R/W	R-only	R-only	R-only	R-only	R-only	R/W
DEBUG_CFG_ COMMIT	Boot	R/W	R-only	R/W	R-only	R-only	R-only	R-only	R-only	R/W
DEBUG_STAT	None	R-only	R-only	R-only	R-only	R-only(1)	R-only	R-only	R-only(1)	R-only
C29DBGEN	Boot	R/W	R-only	R/W	R-only	R-only	R-only	R-only	R-only	R/W
ZONE_DBGEN	None	R-only	R-only	R/W	R-only	R-only	R-only	R-only	R-only	R/W
FLSEMSTAT	None	R-only	R-only	R-only	R-only	R-only(1)	R-only	R-only	R-only(1)	R-only
FLSEMREQ	None	R/W	R/W	R/W	R/W	R-only(1)	R/W	R/W	R-only(1)	R/W
FLSEMCLR	None	R/W(5)	R/W(5)	R/W(5)	R/W(5)	R-only(1)	R/W(5)	R/W(5)	R-only(1)	R/W(5)
BEPROT_BANK	None	R/W(3)	R/W(3)	R/W(3)	R/W(3)	R-only(1)	R/W(3)	R/W(3)	R-only(1)	R/W(3)
BEPROT_STAT	None	R-only	R-only	R-only	R-only	R-only(1)	R-only	R-only	R-only(1)	R-only
BEPROTA	None	R-only	R-only	R-only	R-only	R-only(1)	R-only	R-only	R-only(1)	R-only
BEPROTB	None	R-only	R-only	R-only	R-only	R-only(1)	R-only	R-only	R-only(1)	R-only
BEPROTC	None	R-only	R-only	R-only	R-only	R-only(1)	R-only	R-only	R-only(1)	R-only
WEPROT_CODE_ BANKS*	None	R/W	R/W	R/W	R-only	R-only(1)	R-only	R-only	R-only(1)	R/W
WEPROT_DATA_ BANKS*	None	R/W	R/W	R/W	R-only	R-only(1)	R-only	R-only	R-only(1)	R/W
WEPROT_FLC1_*	None	R/W	R/W	R/W	R-only	R-only(1)	R-only	R-only	R-only(1)	R/W
WEPROT_FLC2_*	None	R/W	R/W	R/W	R-only	R-only(1)	R-only	R-only	R-only(1)	R/W

(1) If any ZONE is enabled for partial or full debug (note that this occurs automatically when SSUMODE = 1 or 2); otherwise, no access.
(2) If ZONE1 is enabled for full debug (note that this occurs automatically when SSUMODE = 1 or 2); otherwise, R-only (with note 1 applied).
(3) Only writable if FLSEMSTAT.ASSIGNED is set and FLSEMSTAT.CPU and FLSEMSTAT.LINK match the transaction's CPUID/LINKID sideband signals.
(4) BANKMAP registers are R/W by CPU1.LINK2 when HSM is in HS-FS state. This allows C29x FW to control updates on non-HSM (or not-secured-yet HSM devices).
(5) FLSEMCLR is only writable if FLSEMSTAT.ASSIGNED is set and FLSEMSTAT.CPU and FLSEMSTAT.LINK match the transaction's CPUID/LINKID, or if the transaction's CPUID.LINKID is CPU1.LINK2.

10.13.2 Permissions for SSU CPU1 Configuration Registers

Table 10-6 describes access permissions to the SSU_CPU1_CFG_REGS register group. This group of registers is not accessible by the HSM.

Table 10-6. Access Permissions for SSU CPU1 Configuration Registers

Register	Register Load Method	Boot	CPU1				CPU2+	
			CPU1 LINK1	CPU1 LINK2	CPU1 LINK3+	CPU1 Debug	CPU2+ LINK0+	CPU2+ Debug
EMU_BOOTDEF_LOW	None	R-only	R-only	R-only	R-only	R/W(1)	No	No
EMU_BOOTDEF_HIGH	None	R-only	R-only	R-only	R-only	R/W(1)	No	No
LINKx_CFG	Boot	R/W	R-only	R-only	R-only	R-only(1)	No	No
STACKy_CFG	Boot	R/W	R-only	R-only	R-only	R-only(1)	No	No
RAMOPENSTAT	None	R-only	R-only	R-only	R-only	R-only(1)	No	No
RAMOPENFRC	Boot	R/W	R/W	R-only	R-only	R/W(1)	No	No
RAMOPENCLR	Boot	R/W	R/W	R-only	R-only	R/W(1)	No	No
DECODER_ADDR_IN	None	R/W	No	R/W	No	No	No	No
DECODER_OUT	None	R-only	No	R-only	No	No	No	No
EMU_DECODER_ADDR_IN	None	No	No	No	No	R/W(1)	No	No
EMU_DECODER_OUT	None	No	No	No	No	R-only(1)	No	No

(1) If ZONE1 is enabled for full debug (note that this occurs automatically when SSUMODE=1 or 2); otherwise, no access.

10.13.3 Permissions for SSU CPU2+ Configuration Registers

Table 10-7 describes the access permissions for the SSU_CPUx_CFG_REGS register groups. This register group exists for each additional CPU present in the device, beginning with CPU2.

Table 10-7. Access Permissions for SSU CPU2/CPU3 Configuration Registers

Register	Boot Load Method	Boot	CPU1			CPUn					Other CPUs
			CPU1 LINK1	CPU1 LINK2+	CPU1 Debug	CPUn LINK0	CPUn LINK1	CPUn LINK2	CPUn LINK3+	CPUn Debug	
RST_VECT	Boot	R/W	R-only	R-only	R-only(1)	R-only	R-only	R-only	R-only	R-only(1)	No
RST_LINK	Boot	R/W	R-only	R-only	R-only(1)	R-only	R-only	R-only	R-only	R-only(1)	No
CPU_RST_CTRL	None	R-only	R-only	R/W	R-only(1)	R-only	R-only	R-only	R-only	R-only(1)	No(3)
DEF_NMI_VECT	Boot	R/W	R-only	R-only	R-only(1)	R-only	R-only	R-only	R-only	R-only(1)	No
DEF_NMI_LINK	Boot	R/W	R-only	R-only	R-only(1)	R-only	R-only	R-only	R-only	R-only(1)	No
EMU_BOOTDEF_LOW	None	R-only	R-only	R-only	R/W(2)	R-only	R-only	R-only	R-only	R/W(2)	No
EMU_BOOTDEF_HIGH	None	R-only	R-only	R-only	R/W(2)	R-only	R-only	R-only	R-only	R/W(2)	No
LINKx_CFG	Boot	R/W	R-only	R-only	R-only(2)	R-only	R-only	R-only	R-only	R-only(2)	No
STACKy_CFG	Boot	R/W	R-only	R-only	R-only(2)	R-only	R-only	R-only	R-only	R-only(2)	No
RAMOPENSTAT	None	R-only	R-only	R-only	R-only(2)	R-only	R-only	R-only	R-only	R-only(2)	No
RAMOPENFRC	Boot	R/W	R/W	R-only	R/W(2)	R/W	R/W	R-only	R-only	R/W(2)	No
RAMOPENCLR	Boot	R/W	R/W	R-only	R/W(2)	R/W	R/W	R-only	R-only	R/W(2)	No
DECODER_ADDR_IN	None	No	No	No	No	R/W	No	R/W	No	No	No
DECODER_OUT	None	No	No	No	No	R-only	No	R-only	No	No	No
EMU_DECODER_ADDR_IN	None	No	No	No	No	No	No	No	No	R/W(2)	No
EMU_DECODER_OUT	None	No	No	No	No	No	No	No	No	R-only(2)	No

(1) If any ZONE is enabled for partial or full debug (note that this occurs automatically when SSUMODE=1 or 2); otherwise, no access.
(2) f ZONE1 is enabled for full debug (note that this occurs automatically when SSUMODE=1 or 2); otherwise, no access.
(3) With exception: the primary CPU's LINK2 has access to modify (R/W) the secondary CPU's CPU_RST_CTRL register. For instance, CPU3.LINK2 has R/W access to CPU_RST_CTRL for CPU4, if present.

10.13.4 Permissions for CPU1 Access Protection Registers

Table 10-8 describes access permissions for the SSU_CPU1_AP_REGS register group. The HSM has no access to these registers.

Table 10-8. Access Permissions for CPU1 Access Protection Registers

Register	Boot Load Method	Boot	CPU1				CPU2+	
			CPU1 LINK1	CPU1 LINK2	CPU1 LINK3+	CPU1 Debug	CPU2+ LINK0+	CPU2+ Debug
APn_CFG	Boot	R/W	R/W(1)	R/W	R-only	R/W(2)	No	No
APn_START[_EXT]	Boot	R/W	R/W(1)	R/W	R-only	R/W(2)	No	No
APn_END[_EXT]	Boot	R/W	R/W(1)	R/W	R-only	R/W(2)	No	No
APn_LOCK	Boot	R/W	R/W(1)	R/W	R-only	R/W(2)	No	No
APn_COMMIT	Boot	R/W	R/W(1)	R/W	R-only	R/W(2)	No	No
APn_ACCESS	Boot	R/W	R/W(1)	R/W	R-only	R/W(2)	No	No

(1) If in SSUMODE1 or SSUMODE2; otherwise, no access.
(2) f ZONE1 is enabled for full debug (note that this occurs automatically when SSUMODE=1 or 2); otherwise, no access.

10.13.5 Permissions for CPU2+ Access Protection Registers

Table 10-9 describes access permissions for the SSU_CPUn_AP_REGS register group or groups, starting with CPU2, for each additional CPU present in a multicore device. The HSM does not have access to these registers.

Table 10-9. Access Permissions for CPU2+ Access Protection Registers

Register	Boot Load Method	Boot	CPU1				CPUn				Other CPUs
			CPU1 LINK1	CPU1 LINK2	CPU1 LINK3+	CPU1 Debug	CPUn LINK0/1	CPUn LINK2	CPUn LINKs	CPUn Debug	
APn_CFG	Boot	R/W	R/W(1)	R/W	No	R/W(2)	R-only	R/W	R-only	R/W(2)	No
APn_START [_EXT]	Boot	R/W	R/W(1)	R/W	No	R/W(2)	R-only	R/W	R-only	R/W(2)	No
APn_END [_EXT]	Boot	R/W	R/W(1)	R/W	No	R/W(2)	R-only	R/W	R-only	R/W(2)	No
APn_LOCK	Boot	R/W	R/W(1)	R/W	No	R/W(2)	R-only	R/W	R-only	R/W(2)	No
APn_ COMMIT	Boot	R/W	R/W(1)	R/W	No	R/W(2)	R-only	R/W	R-only	R/W(2)	No
APn_ ACCESS	Boot	R/W	R/W(1)	R/W	No	R/W(2)	R-only	R/W	R-only	R/W(2)	No

(1) If in SSUMODE1 or SSUMODE2; otherwise, no access.
(2) If ZONE1 is enabled for full debug (note that this occurs automatically when SSUMODE=1 or 2); otherwise, no access.

10.14 SSU Fault Signals

The Safety and Security Unit can generate faults to two different interfaces: the C29x CPU interface, and the Error Aggregator interface that forwards error signals to the Error Signaling Module (ESM). In general, errors that occur as a result of an unauthorized CPU access to a protected memory region generate a fault signal to the CPU, while other types of errors generate a fault signal to the Error Aggregator. The following tables describe the various faults that are generated by the SSU to the CPU and Error Aggregator interfaces. Because the SSU interfaces between the CPU and Flash Controllers for performing program and erase operations, the SSU also reports errors generated by the Flash Controllers to the Error Aggregator. These errors are described in Table 10-11 and Table 10-12.

Table 10-10. SSU to C29x CPU Fault Signals

Security Fault	Error Type	Bus	Error Description
CPUx_PR_HIGHPRIO_ERROR	0x01	Program Bus	This error is generated when a program read occurs, and the address is not covered by a hardcoded protection or a currently active RAMOPEN region, and one of the following is true: • Zero or more than one AP regions respond, or • The responding AP region's XE bit is set to 0, or • The responding AP region's LINKID field is set to 0, or • The responding AP region's LINKID decodes to STACK0 or ZONE0
CPUx_DR1_HIGHPRIO_ERROR	0x01	Data Read Bus 1	This error is generated when a CPU data read occurs on DRB1, and zero or more than one AP region responds with valid read access permissions for the instruction's LINK.
CPUx_DR2_HIGHPRIO_ERROR	0x01	Data Read Bus 2	This error is generated when a CPU data read occurs on DRB2, and zero or more than one AP region responds with valid read access permissions for the instruction's LINK.
CPUx_DW_HIGHPRIOR_ERROR	0x01	Data Write Bus	This error is generated when a CPU data write occurs, and zero or more than one AP region responds with valid write access permissions for the instruction's LINK.

Table 10-11. SSU to Error Aggregator Fault Signals

Error Signal	Description	Error Address	Priority
SSU_ERROR_TYPE[0]	HSM SSU Register Access Error	SSU register address	High
SSU_ERROR_TYPE[1]	CPU1 SSU Register Access Error	SSU register address	High
SSU_ERROR_TYPE[2]	CPU2 SSU Register Access Error	SSU register address	High
SSU_ERROR_TYPE[3]	CPU3 SSU Register Access Error	SSU register address	High
SSU_ERROR_TYPE[4]	CPU4 SSU Register Access Error	SSU register address	High
SSU_ERROR_TYPE[6:5]	Reserved	-	-
SSU_ERROR_TYPE[7]	Invalid BANKMAP, SECVALID, BANKMODE or SSUMODE	N/A	High
SSU_ERROR_TYPE[8]	Error on Flash Controller 1	See Table 10-12	High
SSU_ERROR_TYPE[11:9]	FLC1 Error Type		High
SSU_ERROR_TYPE[12]	FLC1 Register Access Error	FLC1 register address	High
SSU_ERROR_TYPE[13]	Error on Flash Controller 2	See Table 10-12	High
SSU_ERROR_TYPE[16:14]	FLC2 Error Type		High
SSU_ERROR_TYPE[17]	FLC2 Register Access Error	FLC2 register address	High

Table 10-12. Flash Controller Error Codes

Error Code	Error Name	Description	Error Address
0x0	ILLADDR	Controller reported an out of range/illegal Flash address, or a security violation.	Flash address
0x1	ILLPROG	An error occurred while trying to execute a programming command.	
0x2	ILLERASE	An error occurred while trying to execute an erase command.	
0x3	ILLRDVER	The Flash Controller failed to completely program or erase all target bits in the Flash within the maximum number of pulses.	
0x4	ILLMODECH	An attempt was made to execute a program or erase command while the Flash was not in READ mode. The erase/program command was aborted.	
0x5	ILLCMD	An illegal command was issued to the Flash Controller.	Command type
0x6	ILLSIZE	An illegal size value was provided for the Flash Controller command.	Command size
0x7	ILLBANKERASE	An error occurred while trying to execute a bank erase command.	Flash address

10.15 Software

10.15.1 SSU Registers to Driverlib Functions

Table 10-13. SSU Registers to Driverlib Functions

File	Driverlib Function
REVISION	
-	
UPP_REVISION	
-	
MODE	
ssu.h	SSU_getSSUMode
LINK2_AP_OVERRIDE	
ssu.h	SSU_enableLink2APOverride
ssu.h	SSU_disableLink2APOverride
BOOTMODE_STAT	
-	
EMU_BOOTPIN_CONFIG	
-	
EMU_BOOT_DIAG	
-	
EMU_BOOT_CLKCFG	
-	
EMU_BOOTEN	
-	
RAMOPEN_LOCK	
-	
RAMOPEN_COMMIT	
-	
CPUID	
ssu.h	SSU_getCPUID
BANKMAP	
-	
BANKMAP_LOCK	
-	
BANKMAP_COMMIT	
-	
BANKMODE	
-	
BANKMODE_LOCK	
-	
BANKMODE_COMMIT	
-	
SECCFG_UPDATE_CFG	
-	
PROG_BANKMODE	
-	
SECVALID	

Table 10-13. SSU Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
SECVALID_LOCK	
-	
SECVALID_COMMIT	
-	
ZONE1_CFG	
-	
ZONE2_CFG	
-	
ZONE3_CFG	
-	
DEBUG_CFG	
-	
DEBUG_CFG_LOCK	
-	
DEBUG_CFG_COMMIT	
-	
DEBUG_STAT	
-	
C29DBGEN	
-	
ZONE_DBGEN	
-	
BEPROT_BANK	
-	
BEPROT_STAT	
-	
BEPROTA	
-	
BEPROTB	
-	
FLSEMSTAT	
ssu.h	SSU_claimFlashSemaphore
ssu.h	SSU_releaseFlashSemaphore
FLSEMREQ	
ssu.h	SSU_claimFlashSemaphore
FLSEMCLR	
ssu.h	SSU_releaseFlashSemaphore
WEPROT_CODE_BANKS	
-	
WEPROT_CODE_BANKS_LOCK	
-	
WEPROT_CODE_BANKS_COMMIT	
-	
WEPROT_DATA_BANKS	

Table 10-13. SSU Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
WEPROT_DATA_BANKS_LOCK	
-	
WEPROT_DATA_BANKS_COMMIT	
-	
WEPROT_FLC1_B0_A	
-	
WEPROT_FLC1_B0_B	
-	
WEPROT_FLC1_B0_LOCK	
-	
WEPROT_FLC1_B0_COMMIT	
-	
WEPROT_FLC1_B2_A	
-	
WEPROT_FLC1_B2_B	
-	
WEPROT_FLC1_B2_LOCK	
-	
WEPROT_FLC1_B2_COMMIT	
-	
WEPROT_FLC2_B0_A	
-	
WEPROT_FLC2_B0_B	
-	
WEPROT_FLC2_B0_LOCK	
-	
WEPROT_FLC2_B0_COMMIT	
-	
WEPROT_FLC2_B2_A	
-	
WEPROT_FLC2_B2_B	
-	
WEPROT_FLC2_B2_LOCK	
-	
WEPROT_FLC2_B2_COMMIT	
-	
RST_VECT	
ssu.h	SSU_configBootAddress
RST_LINK	
ssu.h	SSU_configBootAddress
CPU_RST_CTRL	
ssu.h	SSU_controlCPUReset
DEF_NMI_VECT	
-	

Table 10-13. SSU Registers to Driverlib Functions (continued)

File	Driverlib Function
DEF_NMI_LINK	
-	
EMU_BOOTDEF_LOW	
-	
EMU_BOOTDEF_HIGH	
-	
LINK3_CFG	
-	
LINK4_CFG	
-	
LINK5_CFG	
-	
LINK6_CFG	
-	
LINK7_CFG	
-	
LINK8_CFG	
-	
LINK9_CFG	
-	
LINK10_CFG	
-	
LINK11_CFG	
-	
LINK12_CFG	
-	
LINK13_CFG	
-	
LINK14_CFG	
-	
LINK15_CFG	
-	
STACK3_CFG	
-	
STACK4_CFG	
-	
STACK5_CFG	
-	
STACK6_CFG	
-	
STACK7_CFG	
-	
RAMOPENSTAT	
ssu.h	SSU_getRAMOPENStatus
RAMOPENFRC	

Table 10-13. SSU Registers to Driverlib Functions (continued)

File	Driverlib Function
ssu.h	SSU_forceRAMOPEN
RAMOPENCLR	
ssu.h	SSU_clearRAMOPEN
DECODER_ADDR_IN	
-	
DECODER_OUT	
-	
EMU_DECODER_ADDR_IN	
-	
EMU_DECODER_OUT	
-	
AP_CFG(i)	
-	
AP_START(i)	
-	
AP_END(i)	
-	
AP_LOCK(i)	
-	
AP_COMMIT(i)	
-	
AP_ACCESS(i)	
-	

10.16 SSU Registers

This section describes the SSU Registers.

10.16.1 SSU Base Address Table

Table 10-14. SSU Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
SSU_GEN_REGS	SSUGEN_BASE	0x3008_0000	-	YES	YES	YES	-	-	YES	YES
SSU_CPU1_CFG_REGS	SSUCPU1CFG_BASE	0x3008_1000	-	YES	-	-	-	-	-	YES
SSU_CPU2_CFG_REGS	SSUCPU2CFG_BASE	0x3008_2000	-	YES	YES	-	-	-	-	YES
SSU_CPU3_CFG_REGS	SSUCPU3CFG_BASE	0x3008_3000	-	YES	-	YES	-	-	-	YES
SSU_CPU1_AP_REGS	SSUCPU1AP_BASE	0x3008_7000	-	YES	-	-	-	-	-	YES
SSU_CPU2_AP_REGS	SSUCPU2AP_BASE	0x3008_8000	-	YES	YES	-	-	-	-	YES
SSU_CPU3_AP_REGS	SSUCPU3AP_BASE	0x3008_9000	-	YES	-	YES	-	-	-	YES

10.16.2 SSU_GEN_REGS Registers

Table 10-15 lists the memory-mapped registers for the SSU_GEN_REGS registers. All register offset addresses not listed in Table 10-15 should be considered as reserved locations and the register contents should not be modified.

Table 10-15. SSU_GEN_REGS Registers

Offset	Acronym	Register Name	Protection
0h	REVISION	Module Revision Register	
4h	UPP_REVISION	User Protection Policy Revision	
8h	SSUMODE	Safety and Security Operational Mode	
Ch	LINK2_AP_OVERRIDE	LINK2 Access Protection Override Register	
2Ch	BOOTMODE_STAT	User Boot Mode Status Register	
30h	EMU_BOOTPIN_CONFIG	User Emulation Boot Pin Configuration	
34h	EMU_BOOT_DIAG	User Emulation Boot Options	
38h	EMU_BOOT_CLKCFG	User Emulation Boot Clock Configuration Register	
3Ch	EMU_BOOTEN	User Emulation Boot Enable	
40h	RAMOPEN_LOCK	RAMOPEN Feature Lock Register	
44h	RAMOPEN_COMMIT	RAMOPEN Feature Commit Register	
4Ch	CPUID	CPUID Register	
50h	BANKMAP	Valid Banks	
54h	BANKMAP_LOCK	Bank Map Lock Register	
58h	BANKMAP_COMMIT	Bank Map Commit Register	
5Ch	BANKMODE	Bank Mode Configuration Register	
60h	BANKMODE_LOCK	Bank Mode Lock Register	
64h	BANKMODE_COMMIT	Bank Mode Commit Register	
68h	SECCFG_UPDATE_CFG	SECCFG Flash Update Configuration Register	
70h	PROG_BANKMODE	Programming BANKMODE Register	
74h	SECVALID	Valid SECCFG Sector	
78h	SECVALID_LOCK	Valid SECCFG Sector Lock Register	
7Ch	SECVALID_COMMIT	Valid SECCFG Sector Commit Register	
80h	ZONE1_CFG	ZONE1 Configuration	
84h	ZONE2_CFG	ZONE2 Configuration	
88h	ZONE3_CFG	ZONE3 Configuration	
90h	DEBUG_CFG	Debug Configuration	
94h	DEBUG_CFG_LOCK	Debug Configuration Lock Register	
98h	DEBUG_CFG_COMMIT	Debug Configuration Commit Register	
A0h	DEBUG_STAT	Debug Status Register	
A4h	C29DBGEN	C29 Debug Enable Register	
A8h	ZONE_DBGEN	ZONE Debug Enable Register	
200h	BEPROT_BANK	Bank Erase Protection Bank Register	
204h	BEPROT_STAT	Bank Erase Status Register	
208h	BEPROTA	Bank Erase Protection Register A	
20Ch	BEPROTB	Bank Erase Protection Register B	
220h	FLSEMSTAT	Flash Controller Semaphore Status Register	
224h	FLSEMREQ	Flash Controller Semaphore Request Register	
228h	FLSEMCLR	Flash Controller Semaphore Clear Register	
230h	WEPROT_CODE_BANKS	Global Code Banks Write Erase	

Table 10-15. SSU_GEN_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
234h	WEPROT_CODE_BANKS_LOCK	Global Code Banks Write Erase Lock	
238h	WEPROT_CODE_BANKS_COMMIT	Global Code Banks Write Erase Commit	
240h	WEPROT_DATA_BANKS	Global Data Banks Write Erase	
244h	WEPROT_DATA_BANKS_LOCK	Global Data Banks Write Erase Lock	
248h	WEPROT_DATA_BANKS_COMMIT	Global Data Banks Write Erase Commit	
300h	WEPROT_FLC1_B0_A	Flash Controller 1 B0 Write Erase Protection A	
304h	WEPROT_FLC1_B0_B	Flash Controller 1 B0 Write Erase Protection B	
310h	WEPROT_FLC1_B0_LOCK	Flash Controller 1 B0 WEPROTA Lock	
314h	WEPROT_FLC1_B0_COMMIT	Flash Controller 1 B0 WEPROTA Commit	
340h	WEPROT_FLC1_B2_A	Flash Controller 1 B2 Write Erase Protection A	
344h	WEPROT_FLC1_B2_B	Flash Controller 1 B2 Write Erase Protection B	
350h	WEPROT_FLC1_B2_LOCK	Flash Controller 1 B2 WEPROTA Lock	
354h	WEPROT_FLC1_B2_COMMIT	Flash Controller 1 B2 WEPROTA Commit	
3A0h	WEPROT_FLC2_B0_A	Flash Controller 2 B0 Write Erase Protection A	
3A4h	WEPROT_FLC2_B0_B	Flash Controller 2 B0 Write Erase Protection B	
3B0h	WEPROT_FLC2_B0_LOCK	Flash Controller 2 B0 WEPROTA Lock	
3B4h	WEPROT_FLC2_B0_COMMIT	Flash Controller 2 B0 WEPROTA Commit	
3E0h	WEPROT_FLC2_B2_A	Flash Controller 2 B2 Write Erase Protection A	
3E4h	WEPROT_FLC2_B2_B	Flash Controller 2 B2 Write Erase Protection B	
3F0h	WEPROT_FLC2_B2_LOCK	Flash Controller 2 B2 WEPROTA Lock	
3F4h	WEPROT_FLC2_B2_COMMIT	Flash Controller 2 B2 WEPROTA Commit	

Complex bit access types are encoded to fit into small table cells. Table 10-16 shows the codes that are used for access types in this section.

Table 10-16. SSU_GEN_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WOnce	W Once	Write Write once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.

Table 10-16. SSU_GEN_REGS Access Type Codes (continued)

Access Type	Code	Description
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

10.16.2.1 REVISION Register (Offset = 0h) [Reset = 00000000h]

REVISION is shown in Figure 10-6 and described in Table 10-17.

Return to the Summary Table.

Module Revision Register

Figure 10-6. REVISION Register

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

Table 10-17. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	RESERVED	R	0h	Reserved
15-8	MAJREV	R	0h	This hardcoded field defines the major revision of the IP. Reset type: XRSn
7-0	MINREV	R	0h	This hardcoded field defines the minor revision of the IP. Reset type: XRSn

10.16.2.2 UPP_REVISION Register (Offset = 4h) [Reset = 00000000h]

UPP_REVISION is shown in Figure 10-7 and described in Table 10-18.

Return to the Summary Table.

User Protection Policy Revision

Figure 10-7. UPP_REVISION Register

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

Table 10-18. UPP_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	REV	R/W	0h	Defines the User Protection Policy revision number stored in the SECCFG sectors. Reset type: XRSn

10.16.2.3 SSUMODE Register (Offset = 8h) [Reset = 00000003h]

SSUMODE is shown in Figure 10-8 and described in Table 10-19.

Return to the Summary Table.

Safety and Security Operational Mode

Figure 10-8. SSUMODE Register

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

Table 10-19. SSUMODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-6	RESERVED	R	0h	Reserved
5-0	OPMODE	R/WOnce	3h	Defines the operational mode of the SSU. 0x03 : SSUMODE3 mode (recommended) 0x0C : SSUMODE2 mode 0x30 : SSUMODE1 mode Others : Invalid and generates a fault. Reset type: XRSn

10.16.2.4 LINK2_AP_OVERRIDE Register (Offset = Ch) [Reset = 00000000h]

LINK2_AP_OVERRIDE is shown in Figure 10-9 and described in Table 10-20.

Return to the Summary Table.

LINK2 Access Protection Override Register

Figure 10-9. LINK2_AP_OVERRIDE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					CPU3	CPU2	CPU1
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 10-20. LINK2_AP_OVERRIDE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	CPU3	R/W	0h	Overrides the effect of every APx_ACCESS.LINK2 bit field for CPU3, providing LINK2 R/W access to each of its enabled AP ranges. The APx_ACCESS register bit fields are not changed. Only CPU3 can modify this bit. 0 : CPU3's APx_ACCESS.LINK2 bit fields are not overridden 1 : CPU3's LINK2 has R/W permissions to all enabled AP ranges. Reset type: XRSn
1	CPU2	R/W	0h	Overrides the effect of every APx_ACCESS.LINK2 bit field for CPU2, providing LINK2 R/W access to each of its enabled AP ranges. The APx_ACCESS register bit fields are not changed. Only CPU2 can modify this bit. 0 : CPU2's APx_ACCESS.LINK2 bit fields are not overridden 1 : CPU2's LINK2 has R/W permissions to all enabled AP ranges. Reset type: XRSn
0	CPU1	R/W	0h	Overrides the effect of every APx_ACCESS.LINK2 bit field for CPU1, providing LINK2 R/W access to each of its enabled AP ranges. The APx_ACCESS register bit fields are not changed. Only CPU1 can modify this bit. 0 : CPU1's APx_ACCESS.LINK2 bit fields are not overridden 1 : CPU1's LINK2 has R/W permissions to all enabled AP ranges. Reset type: XRSn

10.16.2.5 BOOTMODE_STAT Register (Offset = 2Ch) [Reset = 00000000h]

BOOTMODE_STAT is shown in Figure 10-10 and described in Table 10-21.

Return to the Summary Table.

User Boot Mode Status Register

Figure 10-10. BOOTMODE_STAT Register

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

Table 10-21. BOOTMODE_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2-0	PIN_STAT	R/WOnce	0h	Once written by CPU1's boot ROM code, reflects the value of the chosen boot pins. 0x0 : BOOT_DEF0 is active 0x1 : BOOT_DEF1 is active 0x2 : BOOT_DEF2 is active 0x3 : BOOT_DEF3 is active 0x4 : BOOT_DEF4 is active 0x5 : BOOT_DEF5 is active 0x6 : BOOT_DEF6 is active 0x7 : BOOT_DEF7 is active Reset type: XRSn

10.16.2.6 EMU_BOOTPIN_CONFIG Register (Offset = 30h) [Reset = 00FFFFFFh]

EMU_BOOTPIN_CONFIG is shown in Figure 10-11 and described in Table 10-22.

Return to the Summary Table.

User Emulation Boot Pin Configuration

Figure 10-11. EMU_BOOTPIN_CONFIG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY								BMSP2								BMSP1								BMSP0							
R/W-0h								R/W-FFh								R/W-FFh								R/W-FFh							

Table 10-22. EMU_BOOTPIN_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R/W	0h	Defines whether the contents in this register are valid or not. 0x5A : Valid Others : Invalid Reset type: XRSn
23-16	BMSP2	R/W	FFh	Set to the GPIO pin to be used during emulation boot (up to 254). 0x00 : GPIO0 0x01 : GPIO1 ... 0xFE : GPIO254 0xFF : Disabled Reset type: XRSn
15-8	BMSP1	R/W	FFh	Set to the GPIO pin to be used during emulation boot (up to 254). 0x00 : GPIO0 0x01 : GPIO1 ... 0xFE : GPIO254 0xFF : Disabled Reset type: XRSn
7-0	BMSP0	R/W	FFh	Set to the GPIO pin to be used during emulation boot (up to 254). 0x00 : GPIO0 0x01 : GPIO1 ... 0xFE : GPIO254 0xFF : Disabled Reset type: XRSn

10.16.2.7 EMU_BOOT_DIAG Register (Offset = 34h) [Reset = 00000000h]

EMU_BOOT_DIAG is shown in Figure 10-12 and described in Table 10-23.

Return to the Summary Table.

User Emulation Boot Options

Figure 10-12. EMU_BOOT_DIAG Register

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

Table 10-23. EMU_BOOT_DIAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5-4	MPOST_EN	R/W	0h	MPOST Enable. Indicates whether the boot ROM should run MPOST or not during emulation boot. 00 : Run MPOST with PLL disabled (10MHz INTOSC) 01 : Run MPOST at PLLCLK 10 : Run MPOST at PLLCLK/2 11 : Do not run PBIST Reset type: XRSn
3-2	LPOST_COVER	R/W	0h	LPOST Coverage. Indicates the emulation boot coverage option for LPOST, if enabled. 00 : Invalid configuration 01 : ≥ 60% coverage (ASIL-B configuration) 10 : ≥ 80% coverage (ASIL-C configuration) 11 : ≥ 90% coverage (ASIL-D configuration) Reset type: XRSn
1-0	LPOST_EN	R/W	0h	LPOST Enable. Indicates whether the boot ROM should run LPOST or not during emulation boot. 00 : LPOST with PLL disabled (10MHz INTOSC) 01 : LPOST with high shift frequency 10 : LPOST with medium shift frequency 11 : Test disabled (default state) Reset type: XRSn

10.16.2.8 EMU_BOOT_CLKCFG Register (Offset = 38h) [Reset = 00000000h]

EMU_BOOT_CLKCFG is shown in Figure 10-13 and described in Table 10-24.

Return to the Summary Table.

User Emulation Boot Clock Configuration Register

Figure 10-13. EMU_BOOT_CLKCFG Register

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

Table 10-24. EMU_BOOT_CLKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	PLLEN	R/W	0h	Notifies the boot ROM to use the above PLL settings during emulation boot. 0xA : PLL settings valid (enabled) Others : PLL settings not valid (PLL disabled) Reset type: XRSn
27-21	RESERVED	R	0h	Reserved
20-16	REFDIV	R/W	0h	Determines the PLL's REFDIV setting used during emulation boot if enabled. Reset type: XRSn
15-13	RESERVED	R	0h	Reserved
12-8	ODIV	R/W	0h	Determines the PLL's ODIV setting used during emulation boot if enabled. Reset type: XRSn
7-0	IMULT	R/W	0h	Determines the PLL's IMULT setting used during emulation boot if enabled. Reset type: XRSn

10.16.2.9 EMU_BOOTEN Register (Offset = 3Ch) [Reset = 00000000h]

EMU_BOOTEN is shown in Figure 10-14 and described in Table 10-25.

Return to the Summary Table.

User Emulation Boot Enable

Figure 10-14. EMU_BOOTEN Register

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

Table 10-25. EMU_BOOTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	ENABLE	R/W	0h	If set and emulator is connected, emulation boot is enabled. Else, normal boot definition is used. This value is loaded from SECCFG during boot. If left unprogrammed in SECCFG, EMU_BOOT will be enabled (a 1 will be loaded during boot), although R/W restrictions still apply to the EMU_BOOT registers. Reset type: XRSn

10.16.2.10 RAMOPEN_LOCK Register (Offset = 40h) [Reset = 0000005Ah]

RAMOPEN_LOCK is shown in Figure 10-15 and described in Table 10-26.

Return to the Summary Table.

RAMOPEN Feature Lock Register

Figure 10-15. RAMOPEN_LOCK Register

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

Table 10-26. RAMOPEN_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-8	RESERVED	R-0	0h	Reserved
7-0	LOCK	R/W	5Ah	Determines whether RAMOPENFRC can be written from non-LINK0 code. 0xA5 : RAMOPENFRC can be written by LINK0, LINK1, and debugger. 0x5A : RAMOPENFRC can only be written by TI secure LINK0 code. 0ther : RAMOPENFRC can only be written by TI secure LINK0 code. 'Other' values should not be programmed into the SECCFG sector in flash. A value of 0xA5 or 0x5A should always be used. RAMOPEN is always initiated by LINK0 to create space for itself and the standard boot ROM stack (LINK1). After initiating RAMOPEN the secure boot ROM code will set this bit field according to the User Protection Policy programmed into the SECCFG sector of flash. This bit can only be modified if RAMOPEN_COMMIT.COMMIT is zero. Reset type: XRSn

10.16.2.11 RAMOPEN_COMMIT Register (Offset = 44h) [Reset = 00000000h]

RAMOPEN_COMMIT is shown in Figure 10-16 and described in Table 10-27.

Return to the Summary Table.

RAMOPEN Feature Commit Register

Figure 10-16. RAMOPEN_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 10-27. RAMOPEN_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the RAMOPEN_LOCK register. This bit cannot be cleared, except by reset. 0 : RAMOPEN_LOCK is modifiable 1 : RAMOPEN_LOCK is committed permanently Reset type: XRSn

10.16.2.12 CPUID Register (Offset = 4Ch) [Reset = 00000000h]

CPUID is shown in Figure 10-17 and described in Table 10-28.

Return to the Summary Table.

CPUID Register

Figure 10-17. CPUID Register

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

Table 10-28. CPUID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4	DBGREAD	R	0h	This field reflects whether the read of the register is from the debug port or not. Note, this field does not affect the CPUID field. 0 : Not a debug read 1 : Debug read Reset type: XRSn
3	RESERVED	R	0h	Reserved
2-0	CPUID	R	0h	This field reflects the identification value of the CPU performing the read. Along with the DBGREAD field, this provides an easy way for generic SW to determine which CPU it is running from. 0x0 : Invalid 0x1 : CPU1 0x2 : CPU2 (if available) 0x3 : CPU3 (if available) 0x4 : CPU4 (if available) 0x5 : CPU5 (if available) 0x6 : CPU6 (if available) 0x7 : Invalid Reset type: XRSn

10.16.2.13 BANKMAP Register (Offset = 50h) [Reset = 0000C9C9h]

BANKMAP is shown in Figure 10-18 and described in Table 10-29.

Return to the Summary Table.

Valid Banks

Figure 10-18. BANKMAP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CPU3SWAP								CPU1SWAP							
R-0h																R/W-C9h								R/W-C9h							

Table 10-29. BANKMAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	CPU3SWAP	R/W	C9h	Defines which banks mapped in CPU3's code space are mapped as primary (active). The SECCFG sectors are analyzed at boot and the answer written into this register. This bit field is only valid in BANKMODE.MODE = 3. 0xC9 : Default mapping CPU3's banks 0x36 : Alternate mapping of CPU3's banks Others : Invalid and generates a fault. Reset type: XRSn
7-0	CPU1SWAP	R/W	C9h	Defines which banks mapped in CPU1's code space are mapped as primary (active). The SECCFG sectors are analyzed at boot and the answer written into this register. This bit field is only valid in BANKMODE.MODE = 1 or 3. 0xC9 : Default mapping CPU1's banks 0x36 : Alternate mapping of CPU1's banks Others : Invalid and generates a fault if an access to flash occurs while in this state. Reset type: XRSn

10.16.2.14 BANKMAP_LOCK Register (Offset = 54h) [Reset = 00000000h]

BANKMAP_LOCK is shown in Figure 10-19 and described in Table 10-30.

Return to the Summary Table.

Bank Map Lock Register

Figure 10-19. BANKMAP_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						LOCK2	LOCK1
R-0h						R/W-0h	R/W-0h

Table 10-30. BANKMAP_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-2	RESERVED	R	0h	Reserved
1	LOCK2	R/W	0h	When set, locks the bits in the BANKMAP.CPU3SWAP bit field (writes will have no effect on them). This bit can only be modified if BANKMAP_COMMIT.COMMIT2 is cleared. 0 : Unlocked 1 : Locked Reset type: XRSn
0	LOCK1	R/W	0h	When set, locks the bits in the BANKMAP.CPU1SWAP bit field (writes will have no effect on them). This bit can only be modified if BANKMAP_COMMIT.COMMIT1 is cleared. 0 : Unlocked 1 : Locked Reset type: XRSn

10.16.2.15 BANKMAP_COMMIT Register (Offset = 58h) [Reset = 00000000h]

BANKMAP_COMMIT is shown in Figure 10-20 and described in Table 10-31.

Return to the Summary Table.

Bank Map Commit Register

Figure 10-20. BANKMAP_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						COMMIT2	COMMIT1
R-0h						R/W1S-0h	R/W1S-0h

Table 10-31. BANKMAP_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-2	RESERVED	R	0h	Reserved
1	COMMIT2	R/W1S	0h	When set, locks the bits in the BANKMAP_LOCK.LOCK2 bit field (writes will have no effect on them). This bit cannot be cleared, except by reset. 0 : BANKMAP_LOCK.LOCK2 is modifiable 1 : BANKMAP_LOCK.LOCK2 is committed permanently Reset type: XRSn
0	COMMIT1	R/W1S	0h	When set, locks the bits in the BANKMAP_LOCK.LOCK1 bit field (writes will have no effect on them). This bit cannot be cleared, except by reset. 0 : BANKMAP_LOCK.LOCK1 is modifiable 1 : BANKMAP_LOCK.LOCK1 is committed permanently Reset type: XRSn

10.16.2.16 BANKMODE Register (Offset = 5Ch) [Reset = 00000003h]

BANKMODE is shown in Figure 10-21 and described in Table 10-32.

Return to the Summary Table.

Bank Mode Configuration Register

Figure 10-21. BANKMODE Register

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

Table 10-32. BANKMODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	RESERVED	R	0h	Reserved
3-0	MODE	R/W	3h	Defines the configuration of executable flash between CPUs. 0x3 : MODE0. All CPU1, no FOTA. All banks are linearly mapped to CPU1's code space and cannot be remapped/swapped. CPU3 can only run from RAM. 0x6 : MODE1. All CPU1, FOTA enabled. All banks mapped as primary banks are mapped to CPU1's code space. FOTA bank swapping is controlled by BANKMAP.CPU1SWAP. Execution cannot occur from banks that are mapped as alternate banks. CPU3 can only run from RAM. 0x9 : MODE2. CPU1/CPU3 split, no FOTA. A portion of the banks are linearly mapped to CPU1's code space and another portion are linearly mapped to CPU3's code space. Banks cannot be remapped/swapped. 0xC: MODE3. CPU1/CPU3 split, FOTA enabled. A portion of the banks mapped as primary banks are mapped to CPU1's code space and another portion to CPU3's. FOTA bank swapping of CPU1's banks is controlled by BANKMAP.CPU1SWAP, while swapping of CPU3's banks is controlled by BANKMAP.CPU3SWAP. Execution cannot occur from banks that are mapped as alternate banks. Others: Invalid and generates a fault. Reset type: XRSn

10.16.2.17 BANKMODE_LOCK Register (Offset = 60h) [Reset = 00000000h]

BANKMODE_LOCK is shown in Figure 10-22 and described in Table 10-33.

Return to the Summary Table.

Bank Mode Lock Register

Figure 10-22. BANKMODE_LOCK Register

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

Table 10-33. BANKMODE_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks the BANKMODE register (writes will have no effect on it). This bit can only be modified if BANKMODE_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: XRSn

10.16.2.18 BANKMODE_COMMIT Register (Offset = 64h) [Reset = 00000000h]

BANKMODE_COMMIT is shown in Figure 10-23 and described in Table 10-34.

Return to the Summary Table.

Bank Mode Commit Register

Figure 10-23. BANKMODE_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 10-34. BANKMODE_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the bits in the BANKMODE_LOCK register (writes will have no effect on it). This bit cannot be cleared, except by reset. 0 : BANKMAP_LOCK is modifiable 1 : BANKMAP_LOCK is committed permanently Reset type: XRSn

10.16.2.19 SECCFG_UPDATE_CFG Register (Offset = 68h) [Reset = 00003002h]

SECCFG_UPDATE_CFG is shown in Figure 10-24 and described in Table 10-35.

Return to the Summary Table.

SECCFG Flash Update Configuration Register

Figure 10-24. SECCFG_UPDATE_CFG Register

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

Table 10-35. SECCFG_UPDATE_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-12	UPDATE_EN	R/W	3h	Enables SECCFG flash updates. 0x3 : SECCFG is not enabled for flash updates (recommended) 0xC : SECCFG is enabled for flash udpates Others : SECCFG is not enabled for flash updates (not recommended) Note that this field has no effect in SSUMODE1 or SSUMODE2 mode. This field has no effect on the ability to update the BANKMGMT sectors. Reset type: XRSn
11	RESERVED	R	0h	Reserved
10-8	CPU_OWNER	R/W	0h	Along with the LINK field, defines the code designated as the device's SECCFG and BANKMGMT sectors updater. 0x0 : CPU-HSM (if available) 0x1 : CPU1 0x2 : CPU2 (if available) ... 0x7 : CPU7 (if available) Reset type: XRSn
7-4	RESERVED	R	0h	Reserved
3-0	LINK_OWNER	R/W	2h	Along with the CPU field, defines the code designated as the device's SECCFG and BANKMGMT sectors updater. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : LINK1 (i.e., unsecure boot loaders) 0x2 : LINK2 (primary user LINK) 0x3 : LINK3 ... Reset type: XRSn

10.16.2.20 PROG_BANKMODE Register (Offset = 70h) [Reset = 00000000h]

PROG_BANKMODE is shown in Figure 10-25 and described in Table 10-36.

Return to the Summary Table.

Programming BANKMODE Register

Figure 10-25. PROG_BANKMODE Register

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

Table 10-36. PROG_BANKMODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-2	RESERVED	R	0h	Reserved
1-0	MODE	R/W	0h	Defines the expected configuration of flash between CPUs during programming. This value is written during programming while in SSUMODE1 and SSUMODE2 to override the active BANKMODE for the purposes of the programming logic. It does not override the actual active BANKMODE mode, but enables programming code that is intended to be run in a different SSUMODE than the programming is performed in. This register is ignored during SSUMODE3. 0x0 : All CPU1, no FOTA. All banks are linearly mapped to CPU1's code space and cannot be remapped/swapped. CPU3 can only run from RAM. 0x1 : All CPU1, FOTA enabled. All banks mapped as primary banks are mapped to CPU1's code space. FOTA bank swapping is controlled by BANKMAP.CPU1SWAP. Execution cannot occur from banks that are mapped as alternate banks. CPU3 can only run from RAM. 0x2 : CPU1/CPU3 split, no FOTA. A portion of the banks are linearly mapped to CPU1's code space and another portion are linearly mapped to CPU3's code space. Banks cannot be remapped/swapped. 0x3: CPU1/CPU3 split, FOTA enabled. A portion of the banks mapped as primary banks are mapped to CPU1's code space and another portion to CPU3's. FOTA bank swapping of CPU1's banks is controlled by BANKMAP.CPU1SWAP, while swapping of CPU3's banks is controlled by BANKMAP.CPU3SWAP. Execution cannot occur from banks that are mapped as alternate banks. Reset type: XRSn

10.16.2.21 SECVALID Register (Offset = 74h) [Reset = 00C900C9h]

SECVALID is shown in Figure 10-26 and described in Table 10-37.

Return to the Summary Table.

Valid SECCFG Sector

Figure 10-26. SECVALID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CPU3								RESERVED								CPU1							
R-0h								R/W-C9h								R-0h								R/W-C9h							

Table 10-37. SECVALID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	CPU3	R/W	C9h	Defines which SECCFG sector is the valid sector for CPU3. The SECCFG sectors are analyzed at boot and the answer written into this register. 0xC9 : BASE addresses are valid 0x36 : ALT addresses are valid Others : Invalid and generates a fault. Reset type: XRSn
15-8	RESERVED	R	0h	Reserved
7-0	CPU1	R/W	C9h	Defines which SECCFG sector is the valid sector for CPU1 and CPU2. The SECCFG sectors are analyzed at boot and the answer written into this register. 0xC9 : BASE addresses are valid 0x36 : ALT addresses are valid Others : Invalid and generates a fault. Reset type: XRSn

10.16.2.22 SECVALID_LOCK Register (Offset = 78h) [Reset = 00000000h]

SECVALID_LOCK is shown in Figure 10-27 and described in Table 10-38.

Return to the Summary Table.

Valid SECCFG Sector Lock Register

Figure 10-27. SECVALID_LOCK Register

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

Table 10-38. SECVALID_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks the SECVALID register (writes will have no effect on it). This bit can only be modified if SEC_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: XRSn

10.16.2.23 SECVALID_COMMIT Register (Offset = 7Ch) [Reset = 00000000h]

SECVALID_COMMIT is shown in Figure 10-28 and described in Table 10-39.

Return to the Summary Table.

Valid SECCFG Sector Commit Register

Figure 10-28. SECVALID_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 10-39. SECVALID_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the bits in the SEC_LOCK register (writes will have no effect on it). This bit cannot be cleared, except by reset. 0 : SEC_LOCK is modifiable 1 : SEC_LOCK is committed permanently Reset type: XRSn

10.16.2.24 ZONE1_CFG Register (Offset = 80h) [Reset = 00003002h]

ZONE1_CFG is shown in Figure 10-29 and described in Table 10-40.

Return to the Summary Table.

ZONE1 Configuration

Figure 10-29. ZONE1_CFG Register

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

Table 10-40. ZONE1_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-12	UPDATE_EN	R/W	3h	Enables ZONE1 flash updates. 0x3 : ZONE1 is not enabled for flash updates (recommended) 0xC : ZONE1 is enabled for flash udpates Others : ZONE1 is not enabled for flash updates (not recommended) Note that this field has no effect in SSUMODE1 or SSUMODE2 mode. Reset type: XRSn
11	RESERVED	R	0h	Reserved
10-8	FWU_CPU	R/W	0h	Along with the LINK field, defines the CPU allowed to update ZONE1's code. 0x0 : CPU-HSM (if available) 0x1 : CPU1 0x2 : CPU2 (if available) ... 0x7 : CPU7 (if available) Note that CPU1.LINK1 is enabled additionally when the SSU is in SSUMODE1 or SSUMODE2 mode. Reset type: XRSn
7-4	RESERVED	R	0h	Reserved
3-0	FWU_LINK	R/W	2h	Along with the CPU field, defines the LINK allowed to update ZONE1's code. 0x0 : LINK0 (reserved for TI secure boot code and HSM, if available) 0x1 : LINK1 (i.e., unsecure boot loaders) 0x2 : LINK2 (primary user LINK) 0x3 : LINK3 ... 0xF : LINK15 (if available) Note that CPU1.LINK1 is enabled additionally when the SSU is in SSUMODE1 or SSUMODE2 mode. Reset type: XRSn

10.16.2.25 ZONE2_CFG Register (Offset = 84h) [Reset = 00003002h]

ZONE2_CFG is shown in Figure 10-30 and described in Table 10-41.

Return to the Summary Table.

ZONE2 Configuration

Figure 10-30. ZONE2_CFG Register

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

Table 10-41. ZONE2_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-12	UPDATE_EN	R/W	3h	Enables ZONE2 flash updates. 0x3 : ZONE2 is not enabled for flash updates (recommended) 0xC : ZONE2 is enabled for flash udpates Others : ZONE2 is not enabled for flash updates (not recommended) Note that this field has no effect in SSUMODE1 or SSUMODE2 mode. Reset type: XRSn
11	RESERVED	R	0h	Reserved
10-8	FWU_CPU	R/W	0h	Along with the LINK field, defines the CPU allowed to update ZONE2's code. 0x0 : CPU-HSM (if available) 0x1 : CPU1 0x2 : CPU2 (if available) ... 0x7 : CPU7 (if available) Note that CPU1.LINK1 is enabled additionally when the SSU is in SSUMODE1 or SSUMODE2 mode. Reset type: XRSn
7-4	RESERVED	R	0h	Reserved
3-0	FWU_LINK	R/W	2h	Along with the CPU field, defines the LINK allowed to update ZONE2's code. 0x0 : LINK0 (reserved for TI secure boot code and HSM, if available) 0x1 : LINK1 (i.e., unsecure boot loaders) 0x2 : LINK2 (primary user LINK) 0x3 : LINK3 ... 0xF : LINK15 (if available) Note that CPU1.LINK1 is enabled additionally when the SSU is in SSUMODE1 or SSUMODE2 mode. Reset type: XRSn

10.16.2.26 ZONE3_CFG Register (Offset = 88h) [Reset = 00003002h]

ZONE3_CFG is shown in Figure 10-31 and described in Table 10-42.

Return to the Summary Table.

ZONE3 Configuration

Figure 10-31. ZONE3_CFG Register

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

Table 10-42. ZONE3_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-12	UPDATE_EN	R/W	3h	Enables ZONE3 flash updates. 0x3 : ZONE3 is not enabled for flash updates (recommended) 0xC : ZONE3 is enabled for flash udpates Others : ZONE3 is not enabled for flash updates (not recommended) Note that this field has no effect in SSUMODE1 or SSUMODE2 mode. Reset type: XRSn
11	RESERVED	R	0h	Reserved
10-8	FWU_CPU	R/W	0h	Along with the LINK field, defines the CPU allowed to update ZONE3's code. 0x0 : CPU-HSM (if available) 0x1 : CPU1 0x2 : CPU2 (if available) ... 0x7 : CPU7 (if available) Note that CPU1.LINK1 is enabled additionally when the SSU is in SSUMODE1 or SSUMODE2 mode. Reset type: XRSn
7-4	RESERVED	R	0h	Reserved
3-0	FWU_LINK	R/W	2h	Along with the CPU field, defines the LINK allowed to update ZONE3's code. 0x0 : LINK0 (reserved for TI secure boot code and HSM, if available) 0x1 : LINK1 (i.e., unsecure boot loaders) 0x2 : LINK2 (primary user LINK) 0x3 : LINK3 ... 0xF : LINK15 (if available) Note that CPU1.LINK1 is enabled additionally when the SSU is in SSUMODE1 or SSUMODE2 mode. Reset type: XRSn

10.16.2.27 DEBUG_CFG Register (Offset = 90h) [Reset = 00004444h]

DEBUG_CFG is shown in Figure 10-32 and described in Table 10-43.

Return to the Summary Table.

Debug Configuration

Figure 10-32. DEBUG_CFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	ZONE3			RESERVED	ZONE2		
R-0h	R/W-4h			R-0h	R/W-4h		
7	6	5	4	3	2	1	0
RESERVED	ZONE1			RESERVED	C29DBGEN_CFG		
R-0h	R/W-4h			R-0h	R/W-4h		

Table 10-43. DEBUG_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RESERVED	R	0h	Reserved
14-12	ZONE3	R/W	4h	Determines whether a password based method or CPU/Debugger communication based method is allowed to unlock C29 debug. 0x0 : Reserved (C29 debug is disabled) 0x1 : CPU based ZONE_DBGEN.ZONE3 is valid) 0x2 : Password based 0x3 : Reserved (C29 debug is disabled) 0x4 : C29 debug is disabled 0x5 : Reserved (C29 debug is disabled) 0x6 : Reserved (C29 debug is disabled) 0x7 : Reserved (C29 debug is disabled) The corresponding field in SECCFG should not be left unprogrammed (0x7) when programming in SSUMODE3. If the intent is to disable debug, the value 0x4 should be used. Reset type: PORESETn
11	RESERVED	R	0h	Reserved
10-8	ZONE2	R/W	4h	Determines whether a password based method or CPU/Debugger communication based method is allowed to unlock C29 debug. 0x0 : Reserved (C29 debug is disabled) 0x1 : CPU based (ZONE_DBGEN.ZONE2 is valid) 0x2 : Password based 0x3 : Reserved (C29 debug is disabled) 0x4 : C29 debug is disabled 0x5 : Reserved (C29 debug is disabled) 0x6 : Reserved (C29 debug is disabled) 0x7 : Reserved (C29 debug is disabled) The corresponding field in SECCFG should not be left unprogrammed (0x7) when programming in SSUMODE3. If the intent is to disable debug, the value 0x4 should be used. Reset type: PORESETn
7	RESERVED	R	0h	Reserved

Table 10-43. DEBUG_CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6-4	ZONE1	R/W	4h	Determines whether a password based method or CPU/Debugger communication based method is allowed to unlock C29 debug. 0x0 : Reserved (C29 debug is disabled) 0x1 : CPU based (ZONE_DBGEN.ZONE1 is valid) 0x2 : Password based 0x3 : Reserved (C29 debug is disabled) 0x4 : C29 debug is disabled 0x5 : Reserved (C29 debug is disabled) 0x6 : Reserved (C29 debug is disabled) 0x7 : Reserved (C29 debug is disabled) The corresponding field in SECCFG should not be left unprogrammed (0x7) when programming in SSUMODE3. If the intent is to disable debug, the value 0x4 should be used. Reset type: PORESETn
3	RESERVED	R	0h	Reserved
2-0	C29DBGEN_CFG	R/W	4h	Determines whether a password based method or CPU/Debugger communication based method is allowed to unlock C29 debug. 0x0 : Reserved (C29 debug is disabled) 0x1 : CPU based (C29DBGEN.ENA is valid) 0x2 : Password based 0x3 : Reserved (C29 debug is disabled) 0x4 : C29 debug is disabled 0x5 : Reserved (C29 debug is disabled) 0x6 : Reserved (C29 debug is disabled) 0x7 : Reserved (C29 debug is disabled) The corresponding field in SECCFG should not be left unprogrammed (0x7) when programming in SSUMODE3. If the intent is to disable debug, the value 0x4 should be used. Reset type: PORESETn

10.16.2.28 DEBUG_CFG_LOCK Register (Offset = 94h) [Reset = 00000000h]

DEBUG_CFG_LOCK is shown in Figure 10-33 and described in Table 10-44.

Return to the Summary Table.

Debug Configuration Lock Register

Figure 10-33. DEBUG_CFG_LOCK Register

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

Table 10-44. DEBUG_CFG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks this the DEBUG_CFG register (writes will have no effect on them). This bit can only be modified if DEBUG_CFG_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: PORESETn

10.16.2.29 DEBUG_CFG_COMMIT Register (Offset = 98h) [Reset = 00000000h]

DEBUG_CFG_COMMIT is shown in Figure 10-34 and described in Table 10-45.

Return to the Summary Table.

Debug Configuration Commit Register

Figure 10-34. DEBUG_CFG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 10-45. DEBUG_CFG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the DEBUG_CFG_LOCK register. This bit cannot be cleared, except by reset. 0 : DEBUG_CFG_LOCK is modifiable 1 : DEBUG_CFG_LOCK is committed permanently Reset type: PORESETn

10.16.2.30 DEBUG_STAT Register (Offset = A0h) [Reset = 00000000h]

DEBUG_STAT is shown in Figure 10-35 and described in Table 10-46.

Return to the Summary Table.

Debug Status Register

Figure 10-35. DEBUG_STAT Register

31	30	29	28	27	26	25	24
C29DBGS	RESERVED	RESERVED	RESERVED				
R-0h	R-0h	R-0h	R-0h				
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
ZONE3		ZONE2		ZONE1		RESERVED	
R-0h		R-0h		R-0h		R-0h	

Table 10-46. DEBUG_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	C29DBGS	R	0h	Reflects the debug state of the C29 CPUs. 0 : Not enabled for debug 1 : Enabled for debug Reset type: XRSn
30	RESERVED	R	0h	Reserved
29	RESERVED	R	0h	Reserved
28-16	RESERVED	R	0h	Reserved
15-8	RESERVED	R	0h	Reserved
7-6	ZONE3	R	0h	Reflects the state of ZONE3 debug. 00 : Not enabled for debug 01 : Enabled for partial debug 1x : Enabled for full debug Reset type: XRSn
5-4	ZONE2	R	0h	Reflects the state of ZONE2 debug. 00 : Not enabled for debug 01 : Enabled for partial debug 1x : Enabled for full debug Reset type: XRSn
3-2	ZONE1	R	0h	Reflects the state of ZONE1 debug. 00 : Not enabled for debug 01 : Enabled for partial debug 1x : Enabled for full debug Reset type: XRSn
1-0	RESERVED	R	0h	Reserved

10.16.2.31 C29DBGEN Register (Offset = A4h) [Reset = 00000005h]

C29DBGEN is shown in Figure 10-36 and described in Table 10-47.

Return to the Summary Table.

C29 Debug Enable Register

Figure 10-36. C29DBGEN Register

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

Table 10-47. C29DBGEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	ENA	R/W	5h	Enables debug connections to the C29 CPUs. 0xA : CPU taps are enabled (or governed by the HSM, if present) Others : CPU taps are disabled This register is only valid if DEBUG_CFG.C29DBGEN_CFG is set to CPU based. Reset type: PORESETn

10.16.2.32 ZONE_DBGEN Register (Offset = A8h) [Reset = 00030303h]

ZONE_DBGEN is shown in Figure 10-37 and described in Table 10-48.

Return to the Summary Table.

ZONE Debug Enable Register

Figure 10-37. ZONE_DBGEN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED		ZONE3					
R-0h		R/W-3h					
15	14	13	12	11	10	9	8
RESERVED		ZONE2					
R-0h		R/W-3h					
7	6	5	4	3	2	1	0
RESERVED		ZONE1					
R-0h		R/W-3h					

Table 10-48. ZONE_DBGEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	Reserved
21-16	ZONE3	R/W	3h	Enables ZONE3 for full or partial debug 0x03 : ZONE3 is not enabled for debug (recommended) 0x0C : ZONE3 is enabled for partial debug 0x30 : ZONE3 is enabled for full debug Others : ZONE3 is not enabled for debug (not recommended) Reset type: PORESETn
15-14	RESERVED	R	0h	Reserved
13-8	ZONE2	R/W	3h	Enables ZONE2 for full or partial debug 0x03 : ZONE2 is not enabled for debug (recommended) 0x0C : ZONE2 is enabled for partial debug 0x30 : ZONE2 is enabled for full debug Others : ZONE2 is not enabled for debug (not recommended) Reset type: PORESETn
7-6	RESERVED	R	0h	Reserved
5-0	ZONE1	R/W	3h	Enables ZONE1 for full or partial debug 0x03 : ZONE1 is not enabled for debug (recommended) 0x0C : ZONE1 is enabled for partial debug 0x30 : ZONE1 is enabled for full debug Others : ZONE1 is not enabled for debug (not recommended) Reset type: PORESETn

10.16.2.33 BEPROT_BANK Register (Offset = 200h) [Reset = 00000007h]

BEPROT_BANK is shown in Figure 10-38 and described in Table 10-49.

Return to the Summary Table.

Bank Erase Protection Bank Register

Figure 10-38. BEPROT_BANK Register

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

Table 10-49. BEPROT_BANK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-3	FLCID	R/W	0h	When in SSUMODE2 or SSUMODE3, only the FLSEM owner can write to this register. The value written defines which Flash Controller the local BANKID value references. When written, it initiates the Bank Erase State Machine. Reset type: XRSn
2-0	BANKID	R/W	7h	When in SSUMODE2 or SSUMODE3, only the FLSEM owner can write to this register to declare the intent of the bank it intends to bank erase. The value written should be the bank ID local to the Flash Controller it exists in. When written, it initiates the Bank Erase State Machine. Reset type: XRSn

10.16.2.34 BEPROT_STAT Register (Offset = 204h) [Reset = 00000000h]

BEPROT_STAT is shown in Figure 10-39 and described in Table 10-50.

Return to the Summary Table.

Bank Erase Status Register

Figure 10-39. BEPROT_STAT Register

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

Table 10-50. BEPROT_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	ACTIVE	R	0h	Set/Cleared by the Bank Erase State Machine. Set when BEPROT_BANK is written. Cleared when READY is set. Reset type: XRSn
0	READY	R	0h	Set by the Bank Erase State Machine. Cleared when BEPROT_BANK is written. Reset type: XRSn

10.16.2.35 BEPROTA Register (Offset = 208h) [Reset = FFFFFFFFh]

BEPROTA is shown in Figure 10-40 and described in Table 10-51.

Return to the Summary Table.

Bank Erase Protection Register A

Figure 10-40. BEPROTA Register

31	30	29	28	27	26	25	24
SECT31	SECT30	SECT29	SECT28	SECT27	SECT26	SECT25	SECT24
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
23	22	21	20	19	18	17	16
SECT23	SECT22	SECT21	SECT20	SECT19	SECT18	SECT17	SECT16
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
15	14	13	12	11	10	9	8
SECT15	SECT14	SECT13	SECT12	SECT11	SECT10	SECT9	SECT8
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
7	6	5	4	3	2	1	0
SECT7	SECT6	SECT5	SECT4	SECT3	SECT2	SECT1	SECT0
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h

Table 10-51. BEPROTA Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SECT31	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
30	SECT30	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
29	SECT29	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
28	SECT28	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
27	SECT27	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
26	SECT26	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
25	SECT25	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn

Table 10-51. BEPROTA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	SECT24	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
23	SECT23	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
22	SECT22	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
21	SECT21	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
20	SECT20	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
19	SECT19	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
18	SECT18	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
17	SECT17	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
16	SECT16	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
15	SECT15	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
14	SECT14	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
13	SECT13	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
12	SECT12	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
11	SECT11	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn

Table 10-51. BEPROTA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SECT10	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
9	SECT9	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
8	SECT8	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
7	SECT7	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
6	SECT6	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
5	SECT5	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
4	SECT4	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
3	SECT3	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
2	SECT2	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
1	SECT1	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn
0	SECT0	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sector 1 : Bank erase not allowed on sector Reset type: XRSn

10.16.2.36 BEPROTB Register (Offset = 20Ch) [Reset = FFFFFFFFh]

BEPROTB is shown in Figure 10-41 and described in Table 10-52.

Return to the Summary Table.

Bank Erase Protection Register B

Figure 10-41. BEPROTB Register

31	30	29	28	27	26	25	24
SECT255_248	SECT247_240	SECT239_232	SECT231_224	SECT223_216	SECT215_208	SECT207_200	SECT199_192
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
23	22	21	20	19	18	17	16
SECT191_184	SECT183_176	SECT175_168	SECT167_160	SECT159_152	SECT151_144	SECT143_136	SECT135_128
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
15	14	13	12	11	10	9	8
SECT127_120	SECT119_112	SECT111_104	SECT103_96	SECT95_88	SECT87_80	SECT79_72	SECT71_64
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
7	6	5	4	3	2	1	0
SECT63_56	SECT55_48	SECT47_40	SECT39_22	RESERVED	RESERVED	RESERVED	RESERVED
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h

Table 10-52. BEPROTB Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SECT255_248	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
30	SECT247_240	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
29	SECT239_232	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
28	SECT231_224	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
27	SECT223_216	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
26	SECT215_208	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
25	SECT207_200	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn

Table 10-52. BEPROTB Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	SECT199_192	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
23	SECT191_184	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
22	SECT183_176	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
21	SECT175_168	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
20	SECT167_160	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
19	SECT159_152	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
18	SECT151_144	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
17	SECT143_136	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
16	SECT135_128	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
15	SECT127_120	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
14	SECT119_112	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
13	SECT111_104	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
12	SECT103_96	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
11	SECT95_88	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn

Table 10-52. BEPROTB Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SECT87_80	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
9	SECT79_72	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
8	SECT71_64	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
7	SECT63_56	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
6	SECT55_48	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
5	SECT47_40	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
4	SECT39_22	R	1h	Sector Bank Erase Protection 0 : Bank erase allowed on sectors 1 : Bank erase not allowed on sectors Reset type: XRSn
3	RESERVED	R	1h	Reserved
2	RESERVED	R	1h	Reserved
1	RESERVED	R	1h	Reserved
0	RESERVED	R	1h	Reserved

10.16.2.37 FLSEMSTAT Register (Offset = 220h) [Reset = 00000000h]

FLSEMSTAT is shown in Figure 10-42 and described in Table 10-53.

Return to the Summary Table.

Flash Controller Semaphore Status Register

Figure 10-42. FLSEMSTAT Register

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

Table 10-53. FLSEMSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ASSIGNED	R	0h	CPU and LINK fields are valid when this bit is set. 0 : CPU and LINK are unassigned and flash controller is not accessible 1 : CPU and LINK values are valid and the flash controller is currently assigned to them Reset type: XRSn
30	MATCH	R	0h	On a read, this bit will reflect whether the reader's CPUID and LINKID matches the ownership of the flash controller (LINKID matches LINK, CPUID matches CPU, and ASSIGNED equals '1'). 0 : Code performing read does not own the flash controller semaphore 1 : Code performing read owns the flash controller semphore. This avoids code from being required to know which CPU and LINK it belongs to when reading the status of the semaphore. Reset type: XRSn
29-16	RESERVED	R	0h	Reserved
15	RESERVED	R	0h	Reserved
14-12	CPU	R	0h	Defines the current CPU owner of the flash controller. 0x0 : CPU-HSM (if available) 0x1 : CPU1 0x2 : CPU2 (if available) ... 0x7 : CPU7 (if available) Reset type: XRSn
11-10	RESERVED	R	0h	Reserved
9-8	ZONE	R	0h	Defines the current ZONE owner of the flash controller. 0x0 : ZONE0 0x1 : ZONE1 0x2 : ZONE2 0x3 : ZONE3 Reset type: XRSn
7-4	RESERVED	R	0h	Reserved

Table 10-53. FLSEMSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	LINK	R	0h	Defines the current LINK owner of the flash controller. 0x0 : LINK0 0x1 : LINK1 0x2 : LINK2 ... 0xF : LINK15 Reset type: XRSn

10.16.2.38 FLSEMREQ Register (Offset = 224h) [Reset = 00000000h]

FLSEMREQ is shown in Figure 10-43 and described in Table 10-54.

Return to the Summary Table.

Flash Controller Semaphore Request Register

Figure 10-43. FLSEMREQ Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															REQ
R-0h															R-0/W1S-0h

Table 10-54. FLSEMREQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	REQ	R-0/W1S	0h	If the FLSEMSTAT.ASSIGNED bit is cleared, writing a '1' to this bit causes the SSU logic to: 1) Load the LINKID of the code performing the write into the FLSEMSTAT.LINK bit field, and 2) Load the ZONEID of the code performing the write into the FLSEMSTAT.ZONE bit field, and 3) Load the CPUID of the code performing the write into the FLSEMSTAT.CPU bit field, and 4) Set the FLSEMSTAT.ASSIGNED bit. If the FLSEMSTAT.ASSIGNED is already set when a write to this bit occurs, the write will be ignored. Code should check the contents of FLSEMSTAT.MATCH after writing to this bit to determine if it was successfully granted ownership. If the above conditions are not met during the write, the write will be ignored with no error indicator. It is advised that the writing code perform a read of the FLSEMSTAT register to ensure it was set after writing to this bit. Note that if the HSM (if present) performs the write, the CPUID, LINKID, and ZONEID values that are captured are all zeros. Reset type: XRSn

10.16.2.39 FLSEMCLR Register (Offset = 228h) [Reset = 00000000h]

FLSEMCLR is shown in Figure 10-44 and described in Table 10-55.

Return to the Summary Table.

Flash Controller Semaphore Clear Register

Figure 10-44. FLSEMCLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															CLR
R-0h															R-0/W1S-0h

Table 10-55. FLSEMCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	CLR	R-0/W1S	0h	If the following conditions are met, a write of '1' to this bit causes the SSU logic to load the FLSEMSTAT register to its reset state: 1) The CPUID.LINKID of the code performing the write is CPU1.LINK2 and FLSEMSTAT.CPU and FLSEMSTAT.LINK do not match the HSM's designed CPUID/LINKID (both zero), or 2) The CPUID and LINKID of the code performing the write is the HSM (both zero), or 3) FLSEMSTAT.ASSIGNED bit is set, and a) FLSEMSTAT.LINK equals the LINKID of the code performing the write, and b) FLSEMSTAT.CPU equals the CPUID of the code performing the write. If the above conditions are not met during the write, the write will be ignored with no error indicator. It is advised that the writing code perform a read of the FLSEMSTAT register to ensure it was cleared after writing to this bit. Reset type: XRSn

10.16.2.40 WEPROT_CODE_BANKS Register (Offset = 230h) [Reset = 00000000h]

WEPROT_CODE_BANKS is shown in Figure 10-45 and described in Table 10-56.

Return to the Summary Table.

Global Code Banks Write Erase

Figure 10-45. WEPROT_CODE_BANKS Register

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

Table 10-56. WEPROT_CODE_BANKS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	PROT	R/W	0h	When set, protects all code banks outside of the HSM from being programmed or erased. Reset type: XRSn

10.16.2.41 WEPROT_CODE_BANKS_LOCK Register (Offset = 234h) [Reset = 00000000h]

WEPROT_CODE_BANKS_LOCK is shown in Figure 10-46 and described in Table 10-57.

Return to the Summary Table.

Global Code Banks Write Erase Lock

Figure 10-46. WEPROT_CODE_BANKS_LOCK Register

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

Table 10-57. WEPROT_CODE_BANKS_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks the bits in the WEPROT_CODE_BANKS register (writes will have no effect on it). This bit can only be modified if WEPROT_CODE_BANKS_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: XRSn

10.16.2.42 WEPROT_CODE_BANKS_COMMIT Register (Offset = 238h) [Reset = 00000000h]

WEPROT_CODE_BANKS_COMMIT is shown in Figure 10-47 and described in Table 10-58.

Return to the Summary Table.

Global Code Banks Write Erase Commit

Figure 10-47. WEPROT_CODE_BANKS_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 10-58. WEPROT_CODE_BANKS_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the bits in the WEPROT_CODE_BANKS_LOCK register (writes will have no effect on it). This bit cannot be cleared, except by reset. 0 : WEPROT_CODE_BANKS_LOCK is modifiable 1 : WEPROT_CODE_BANKS_LOCK is committed permanently Reset type: XRSn

10.16.2.43 WEPROT_DATA_BANKS Register (Offset = 240h) [Reset = 00000000h]

WEPROT_DATA_BANKS is shown in Figure 10-48 and described in Table 10-59.

Return to the Summary Table.

Global Data Banks Write Erase

Figure 10-48. WEPROT_DATA_BANKS Register

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

Table 10-59. WEPROT_DATA_BANKS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	PROT	R/W	0h	When set, protects all data banks from being programmed or erased. Reset type: XRSn

10.16.2.44 WEPROT_DATA_BANKS_LOCK Register (Offset = 244h) [Reset = 00000000h]

WEPROT_DATA_BANKS_LOCK is shown in Figure 10-49 and described in Table 10-60.

Return to the Summary Table.

Global Data Banks Write Erase Lock

Figure 10-49. WEPROT_DATA_BANKS_LOCK Register

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

Table 10-60. WEPROT_DATA_BANKS_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks the bits in the WEPROT_DATA_BANKS register (writes will have no effect on it). This bit can only be modified if WEPROT_DATA_BANKS_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: XRSn

10.16.2.45 WEPROT_DATA_BANKS_COMMIT Register (Offset = 248h) [Reset = 00000000h]

WEPROT_DATA_BANKS_COMMIT is shown in Figure 10-50 and described in Table 10-61.

Return to the Summary Table.

Global Data Banks Write Erase Commit

Figure 10-50. WEPROT_DATA_BANKS_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 10-61. WEPROT_DATA_BANKS_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the bits in the WEPROT_DATA_BANKS_LOCK register (writes will have no effect on it). This bit cannot be cleared, except by reset. 0 : WEPROT_DATA_BANKS_LOCK is modifiable 1 : WEPROT_DATA_BANKS_LOCK is committed permanently Reset type: XRSn

10.16.2.46 WEPROT_FLC1_B0_A Register (Offset = 300h) [Reset = FFFFFFFFh]

WEPROT_FLC1_B0_A is shown in Figure 10-51 and described in Table 10-62.

Return to the Summary Table.

Flash Controller 1 B0 Write Erase Protection A

Figure 10-51. WEPROT_FLC1_B0_A Register

31	30	29	28	27	26	25	24
SECT31	SECT30	SECT29	SECT28	SECT27	SECT26	SECT25	SECT24
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
SECT23	SECT22	SECT21	SECT20	SECT19	SECT18	SECT17	SECT16
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
SECT15	SECT14	SECT13	SECT12	SECT11	SECT10	SECT9	SECT8
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
SECT7	SECT6	SECT5	SECT4	SECT3	SECT2	SECT1	SECT0
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 10-62. WEPROT_FLC1_B0_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SECT31	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
30	SECT30	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
29	SECT29	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
28	SECT28	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
27	SECT27	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
26	SECT26	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
25	SECT25	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn

Table 10-62. WEPROT_FLC1_B0_A Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	SECT24	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
23	SECT23	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
22	SECT22	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
21	SECT21	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
20	SECT20	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
19	SECT19	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
18	SECT18	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
17	SECT17	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
16	SECT16	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
15	SECT15	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
14	SECT14	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
13	SECT13	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
12	SECT12	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
11	SECT11	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn

Table 10-62. WEPROT_FLC1_B0_A Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SECT10	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
9	SECT9	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
8	SECT8	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
7	SECT7	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
6	SECT6	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
5	SECT5	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
4	SECT4	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
3	SECT3	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
2	SECT2	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
1	SECT1	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
0	SECT0	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn

10.16.2.47 WEPROT_FLC1_B0_B Register (Offset = 304h) [Reset = FFFFFFFFh]

WEPROT_FLC1_B0_B is shown in Figure 10-52 and described in Table 10-63.

Return to the Summary Table.

Flash Controller 1 B0 Write Erase Protection B

Figure 10-52. WEPROT_FLC1_B0_B Register

31	30	29	28	27	26	25	24
SECT255_248	SECT247_240	SECT239_232	SECT231_224	SECT223_216	SECT215_208	SECT207_200	SECT199_192
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
SECT191_184	SECT183_176	SECT175_168	SECT167_160	SECT159_152	SECT151_144	SECT143_136	SECT135_128
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
SECT127_120	SECT119_112	SECT111_104	SECT103_96	SECT95_88	SECT87_80	SECT79_72	SECT71_64
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
SECT63_56	SECT55_48	SECT47_40	SECT39_22	RESERVED	RESERVED	RESERVED	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R-1h	R-1h	R-1h	R-1h

Table 10-63. WEPROT_FLC1_B0_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SECT255_248	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
30	SECT247_240	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
29	SECT239_232	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
28	SECT231_224	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
27	SECT223_216	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
26	SECT215_208	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
25	SECT207_200	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn

Table 10-63. WEPROT_FLC1_B0_B Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	SECT199_192	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
23	SECT191_184	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
22	SECT183_176	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
21	SECT175_168	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
20	SECT167_160	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
19	SECT159_152	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
18	SECT151_144	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
17	SECT143_136	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
16	SECT135_128	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
15	SECT127_120	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
14	SECT119_112	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
13	SECT111_104	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
12	SECT103_96	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
11	SECT95_88	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn

Table 10-63. WEPROT_FLC1_B0_B Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SECT87_80	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
9	SECT79_72	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
8	SECT71_64	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
7	SECT63_56	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
6	SECT55_48	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
5	SECT47_40	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
4	SECT39_22	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
3	RESERVED	R	1h	Reserved
2	RESERVED	R	1h	Reserved
1	RESERVED	R	1h	Reserved
0	RESERVED	R	1h	Reserved

10.16.2.48 WEPROT_FLC1_B0_LOCK Register (Offset = 310h) [Reset = 00000000h]

WEPROT_FLC1_B0_LOCK is shown in Figure 10-53 and described in Table 10-64.

Return to the Summary Table.

Flash Controller 1 B0 WEPROTA Lock

Figure 10-53. WEPROT_FLC1_B0_LOCK Register

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

Table 10-64. WEPROT_FLC1_B0_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks the bits in the WEPROT_FLC1_B0* registers (writes will have no effect on them). This bit can only be modified if WEPROT_FLC1_B0_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: XRSn

10.16.2.49 WEPROT_FLC1_B0_COMMIT Register (Offset = 314h) [Reset = 00000000h]

WEPROT_FLC1_B0_COMMIT is shown in Figure 10-54 and described in Table 10-65.

Return to the Summary Table.

Flash Controller 1 B0 WEPROTA Commit

Figure 10-54. WEPROT_FLC1_B0_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 10-65. WEPROT_FLC1_B0_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the bits in the WEPROT_FLC1_B0_LOCK register (writes will have no effect on it). This bit cannot be cleared, except by reset. 0 : WEPROT_FLC1_B0_LOCK is modifiable 1 : WEPROT_FLC1_B0_LOCK is committed permanently Reset type: XRSn

10.16.2.50 WEPROT_FLC1_B2_A Register (Offset = 340h) [Reset = FFFFFFFFh]

WEPROT_FLC1_B2_A is shown in Figure 10-55 and described in Table 10-66.

Return to the Summary Table.

Flash Controller 1 B2 Write Erase Protection A

Figure 10-55. WEPROT_FLC1_B2_A Register

31	30	29	28	27	26	25	24
SECT31	SECT30	SECT29	SECT28	SECT27	SECT26	SECT25	SECT24
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
SECT23	SECT22	SECT21	SECT20	SECT19	SECT18	SECT17	SECT16
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
SECT15	SECT14	SECT13	SECT12	SECT11	SECT10	SECT9	SECT8
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
SECT7	SECT6	SECT5	SECT4	SECT3	SECT2	SECT1	SECT0
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 10-66. WEPROT_FLC1_B2_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SECT31	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
30	SECT30	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
29	SECT29	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
28	SECT28	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
27	SECT27	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
26	SECT26	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
25	SECT25	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn

Table 10-66. WEPROT_FLC1_B2_A Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	SECT24	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
23	SECT23	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
22	SECT22	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
21	SECT21	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
20	SECT20	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
19	SECT19	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
18	SECT18	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
17	SECT17	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
16	SECT16	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
15	SECT15	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
14	SECT14	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
13	SECT13	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
12	SECT12	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
11	SECT11	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn

Table 10-66. WEPROT_FLC1_B2_A Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SECT10	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
9	SECT9	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
8	SECT8	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
7	SECT7	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
6	SECT6	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
5	SECT5	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
4	SECT4	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
3	SECT3	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
2	SECT2	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
1	SECT1	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
0	SECT0	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn

10.16.2.51 WEPROT_FLC1_B2_B Register (Offset = 344h) [Reset = FFFFFFFFh]

WEPROT_FLC1_B2_B is shown in Figure 10-56 and described in Table 10-67.

Return to the Summary Table.

Flash Controller 1 B2 Write Erase Protection B

Figure 10-56. WEPROT_FLC1_B2_B Register

31	30	29	28	27	26	25	24
SECT255_248	SECT247_240	SECT239_232	SECT231_224	SECT223_216	SECT215_208	SECT207_200	SECT199_192
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
SECT191_184	SECT183_176	SECT175_168	SECT167_160	SECT159_152	SECT151_144	SECT143_136	SECT135_128
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
SECT127_120	SECT119_112	SECT111_104	SECT103_96	SECT95_88	SECT87_80	SECT79_72	SECT71_64
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
SECT63_56	SECT55_48	SECT47_40	SECT39_22	RESERVED	RESERVED	RESERVED	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R-1h	R-1h	R-1h	R-1h

Table 10-67. WEPROT_FLC1_B2_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SECT255_248	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
30	SECT247_240	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
29	SECT239_232	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
28	SECT231_224	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
27	SECT223_216	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
26	SECT215_208	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
25	SECT207_200	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn

Table 10-67. WEPROT_FLC1_B2_B Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	SECT199_192	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
23	SECT191_184	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
22	SECT183_176	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
21	SECT175_168	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
20	SECT167_160	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
19	SECT159_152	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
18	SECT151_144	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
17	SECT143_136	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
16	SECT135_128	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
15	SECT127_120	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
14	SECT119_112	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
13	SECT111_104	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
12	SECT103_96	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
11	SECT95_88	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn

Table 10-67. WEPROT_FLC1_B2_B Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SECT87_80	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
9	SECT79_72	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
8	SECT71_64	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
7	SECT63_56	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
6	SECT55_48	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
5	SECT47_40	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
4	SECT39_22	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
3	RESERVED	R	1h	Reserved
2	RESERVED	R	1h	Reserved
1	RESERVED	R	1h	Reserved
0	RESERVED	R	1h	Reserved

10.16.2.52 WEPROT_FLC1_B2_LOCK Register (Offset = 350h) [Reset = 00000000h]

WEPROT_FLC1_B2_LOCK is shown in Figure 10-57 and described in Table 10-68.

Return to the Summary Table.

Flash Controller 1 B2 WEPROTA Lock

Figure 10-57. WEPROT_FLC1_B2_LOCK Register

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

Table 10-68. WEPROT_FLC1_B2_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks the bits in the WEPROT_FLC1_B2* registers (writes will have no effect on them). This bit can only be modified if WEPROT_FLC1_B2_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: XRSn

10.16.2.53 WEPROT_FLC1_B2_COMMIT Register (Offset = 354h) [Reset = 00000000h]

WEPROT_FLC1_B2_COMMIT is shown in Figure 10-58 and described in Table 10-69.

Return to the Summary Table.

Flash Controller 1 B2 WEPROTA Commit

Figure 10-58. WEPROT_FLC1_B2_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 10-69. WEPROT_FLC1_B2_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the bits in the WEPROT_FLC1_B2_LOCK register (writes will have no effect on it). This bit cannot be cleared, except by reset. 0 : WEPROT_FLC1_B2_LOCK is modifiable 1 : WEPROT_FLC1_B2_LOCK is committed permanently Reset type: XRSn

10.16.2.54 WEPROT_FLC2_B0_A Register (Offset = 3A0h) [Reset = FFFFFFFFh]

WEPROT_FLC2_B0_A is shown in Figure 10-59 and described in Table 10-70.

Return to the Summary Table.

Flash Controller 2 B0 Write Erase Protection A

Figure 10-59. WEPROT_FLC2_B0_A Register

31	30	29	28	27	26	25	24
SECT31	SECT30	SECT29	SECT28	SECT27	SECT26	SECT25	SECT24
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
SECT23	SECT22	SECT21	SECT20	SECT19	SECT18	SECT17	SECT16
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
SECT15	SECT14	SECT13	SECT12	SECT11	SECT10	SECT9	SECT8
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
SECT7	SECT6	SECT5	SECT4	SECT3	SECT2	SECT1	SECT0
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 10-70. WEPROT_FLC2_B0_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SECT31	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
30	SECT30	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
29	SECT29	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
28	SECT28	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
27	SECT27	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
26	SECT26	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
25	SECT25	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn

Table 10-70. WEPROT_FLC2_B0_A Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	SECT24	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
23	SECT23	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
22	SECT22	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
21	SECT21	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
20	SECT20	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
19	SECT19	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
18	SECT18	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
17	SECT17	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
16	SECT16	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
15	SECT15	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
14	SECT14	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
13	SECT13	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
12	SECT12	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
11	SECT11	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn

Table 10-70. WEPROT_FLC2_B0_A Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SECT10	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
9	SECT9	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
8	SECT8	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
7	SECT7	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
6	SECT6	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
5	SECT5	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
4	SECT4	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
3	SECT3	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
2	SECT2	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
1	SECT1	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
0	SECT0	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn

10.16.2.55 WEPROT_FLC2_B0_B Register (Offset = 3A4h) [Reset = FFFFFFFFh]

WEPROT_FLC2_B0_B is shown in Figure 10-60 and described in Table 10-71.

Return to the Summary Table.

Flash Controller 2 B0 Write Erase Protection B

Figure 10-60. WEPROT_FLC2_B0_B Register

31	30	29	28	27	26	25	24
SECT255_248	SECT247_240	SECT239_232	SECT231_224	SECT223_216	SECT215_208	SECT207_200	SECT199_192
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
SECT191_184	SECT183_176	SECT175_168	SECT167_160	SECT159_152	SECT151_144	SECT143_136	SECT135_128
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
SECT127_120	SECT119_112	SECT111_104	SECT103_96	SECT95_88	SECT87_80	SECT79_72	SECT71_64
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
SECT63_56	SECT55_48	SECT47_40	SECT39_22	RESERVED	RESERVED	RESERVED	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R-1h	R-1h	R-1h	R-1h

Table 10-71. WEPROT_FLC2_B0_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SECT255_248	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
30	SECT247_240	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
29	SECT239_232	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
28	SECT231_224	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
27	SECT223_216	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
26	SECT215_208	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
25	SECT207_200	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn

Table 10-71. WEPROT_FLC2_B0_B Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	SECT199_192	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
23	SECT191_184	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
22	SECT183_176	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
21	SECT175_168	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
20	SECT167_160	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
19	SECT159_152	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
18	SECT151_144	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
17	SECT143_136	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
16	SECT135_128	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
15	SECT127_120	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
14	SECT119_112	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
13	SECT111_104	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
12	SECT103_96	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
11	SECT95_88	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn

Table 10-71. WEPROT_FLC2_B0_B Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SECT87_80	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
9	SECT79_72	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
8	SECT71_64	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
7	SECT63_56	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
6	SECT55_48	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
5	SECT47_40	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
4	SECT39_22	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
3	RESERVED	R	1h	Reserved
2	RESERVED	R	1h	Reserved
1	RESERVED	R	1h	Reserved
0	RESERVED	R	1h	Reserved

10.16.2.56 WEPROT_FLC2_B0_LOCK Register (Offset = 3B0h) [Reset = 00000000h]

WEPROT_FLC2_B0_LOCK is shown in Figure 10-61 and described in Table 10-72.

Return to the Summary Table.

Flash Controller 2 B0 WEPROTA Lock

Figure 10-61. WEPROT_FLC2_B0_LOCK Register

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

Table 10-72. WEPROT_FLC2_B0_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks the bits in the WEPROT_FLC2_B0* registers (writes will have no effect on them). This bit can only be modified if WEPROT_FLC2_B0_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: XRSn

10.16.2.57 WEPROT_FLC2_B0_COMMIT Register (Offset = 3B4h) [Reset = 00000000h]

WEPROT_FLC2_B0_COMMIT is shown in Figure 10-62 and described in Table 10-73.

Return to the Summary Table.

Flash Controller 2 B0 WEPROTA Commit

Figure 10-62. WEPROT_FLC2_B0_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 10-73. WEPROT_FLC2_B0_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the bits in the WEPROT_FLC2_B0_LOCK register (writes will have no effect on it). This bit cannot be cleared, except by reset. 0 : WEPROT_FLC2_B0_LOCK is modifiable 1 : WEPROT_FLC2_B0_LOCK is committed permanently Reset type: XRSn

10.16.2.58 WEPROT_FLC2_B2_A Register (Offset = 3E0h) [Reset = FFFFFFFFh]

WEPROT_FLC2_B2_A is shown in Figure 10-63 and described in Table 10-74.

Return to the Summary Table.

Flash Controller 2 B2 Write Erase Protection A

Figure 10-63. WEPROT_FLC2_B2_A Register

31	30	29	28	27	26	25	24
SECT31	SECT30	SECT29	SECT28	SECT27	SECT26	SECT25	SECT24
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
SECT23	SECT22	SECT21	SECT20	SECT19	SECT18	SECT17	SECT16
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
SECT15	SECT14	SECT13	SECT12	SECT11	SECT10	SECT9	SECT8
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
SECT7	SECT6	SECT5	SECT4	SECT3	SECT2	SECT1	SECT0
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 10-74. WEPROT_FLC2_B2_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SECT31	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
30	SECT30	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
29	SECT29	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
28	SECT28	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
27	SECT27	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
26	SECT26	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
25	SECT25	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn

Table 10-74. WEPROT_FLC2_B2_A Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	SECT24	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
23	SECT23	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
22	SECT22	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
21	SECT21	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
20	SECT20	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
19	SECT19	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
18	SECT18	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
17	SECT17	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
16	SECT16	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
15	SECT15	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
14	SECT14	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
13	SECT13	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
12	SECT12	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
11	SECT11	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn

Table 10-74. WEPROT_FLC2_B2_A Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SECT10	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
9	SECT9	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
8	SECT8	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
7	SECT7	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
6	SECT6	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
5	SECT5	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
4	SECT4	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
3	SECT3	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
2	SECT2	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
1	SECT1	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn
0	SECT0	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sector 1 : Write erase not allowed on sector Reset type: XRSn

10.16.2.59 WEPROT_FLC2_B2_B Register (Offset = 3E4h) [Reset = FFFFFFFFh]

WEPROT_FLC2_B2_B is shown in Figure 10-64 and described in Table 10-75.

Return to the Summary Table.

Flash Controller 2 B2 Write Erase Protection B

Figure 10-64. WEPROT_FLC2_B2_B Register

31	30	29	28	27	26	25	24
SECT255_248	SECT247_240	SECT239_232	SECT231_224	SECT223_216	SECT215_208	SECT207_200	SECT199_192
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
SECT191_184	SECT183_176	SECT175_168	SECT167_160	SECT159_152	SECT151_144	SECT143_136	SECT135_128
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
SECT127_120	SECT119_112	SECT111_104	SECT103_96	SECT95_88	SECT87_80	SECT79_72	SECT71_64
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
SECT63_56	SECT55_48	SECT47_40	SECT39_22	RESERVED	RESERVED	RESERVED	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R-1h	R-1h	R-1h	R-1h

Table 10-75. WEPROT_FLC2_B2_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SECT255_248	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
30	SECT247_240	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
29	SECT239_232	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
28	SECT231_224	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
27	SECT223_216	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
26	SECT215_208	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
25	SECT207_200	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn

Table 10-75. WEPROT_FLC2_B2_B Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	SECT199_192	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
23	SECT191_184	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
22	SECT183_176	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
21	SECT175_168	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
20	SECT167_160	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
19	SECT159_152	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
18	SECT151_144	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
17	SECT143_136	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
16	SECT135_128	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
15	SECT127_120	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
14	SECT119_112	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
13	SECT111_104	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
12	SECT103_96	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
11	SECT95_88	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn

Table 10-75. WEPROT_FLC2_B2_B Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SECT87_80	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
9	SECT79_72	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
8	SECT71_64	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
7	SECT63_56	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
6	SECT55_48	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
5	SECT47_40	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
4	SECT39_22	R/W	1h	Sector Write Erase Protection 0 : Write erase allowed on sectors 1 : Write erase not allowed on sectors Reset type: XRSn
3	RESERVED	R	1h	Reserved
2	RESERVED	R	1h	Reserved
1	RESERVED	R	1h	Reserved
0	RESERVED	R	1h	Reserved

10.16.2.60 WEPROT_FLC2_B2_LOCK Register (Offset = 3F0h) [Reset = 00000000h]

WEPROT_FLC2_B2_LOCK is shown in Figure 10-65 and described in Table 10-76.

Return to the Summary Table.

Flash Controller 2 B2 WEPROTA Lock

Figure 10-65. WEPROT_FLC2_B2_LOCK Register

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

Table 10-76. WEPROT_FLC2_B2_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks the bits in the WEPROT_FLC2_B2* registers (writes will have no effect on them). This bit can only be modified if WEPROT_FLC2_B2_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: XRSn

10.16.2.61 WEPROT_FLC2_B2_COMMIT Register (Offset = 3F4h) [Reset = 00000000h]

WEPROT_FLC2_B2_COMMIT is shown in Figure 10-66 and described in Table 10-77.

Return to the Summary Table.

Flash Controller 2 B2 WEPROTA Commit

Figure 10-66. WEPROT_FLC2_B2_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 10-77. WEPROT_FLC2_B2_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the bits in the WEPROT_FLC2_B2_LOCK register (writes will have no effect on it). This bit cannot be cleared, except by reset. 0 : WEPROT_FLC2_B2_LOCK is modifiable 1 : WEPROT_FLC2_B2_LOCK is committed permanently Reset type: XRSn

10.16.3 SSU_CPU1_CFG_REGS Registers

Table 10-78 lists the memory-mapped registers for the SSU_CPU1_CFG_REGS registers. All register offset addresses not listed in Table 10-78 should be considered as reserved locations and the register contents should not be modified.

Table 10-78. SSU_CPU1_CFG_REGS Registers

Offset	Acronym	Register Name	Protection
28h	EMU_BOOTDEF_LOW	User Emulation Boot Definition Low Register	
2Ch	EMU_BOOTDEF_HIGH	User Emulation Boot Definition High Register	
3Ch	LINK3_CFG	LINK3 Configuration	
40h	LINK4_CFG	LINK4 Configuration	
44h	LINK5_CFG	LINK5 Configuration	
48h	LINK6_CFG	LINK6 Configuration	
4Ch	LINK7_CFG	LINK7 Configuration	
50h	LINK8_CFG	LINK8 Configuration	
54h	LINK9_CFG	LINK9 Configuration	
58h	LINK10_CFG	LINK10 Configuration	
5Ch	LINK11_CFG	LINK11 Configuration	
60h	LINK12_CFG	LINK12 Configuration	
64h	LINK13_CFG	LINK13 Configuration	
68h	LINK14_CFG	LINK14 Configuration	
6Ch	LINK15_CFG	LINK15 Configuration	
7Ch	STACK3_CFG	STACK3 Configuration	
80h	STACK4_CFG	STACK4 Configuration	
84h	STACK5_CFG	STACK5 Configuration	
88h	STACK6_CFG	STACK6 Configuration	
8Ch	STACK7_CFG	STACK7 Configuration	
90h	RAMOPENSTAT	RAMOPEN Feature Status Register	
94h	RAMOPENFRC	RAMOPEN Feature Force Register	
98h	RAMOPENCLR	RAMOPEN Feature Clear Register	
A0h	DECODER_ADDR_IN	SW ZONE Decoder Address Input	
A4h	DECODER_OUT	SW ZONE Decoder Output	
A8h	EMU_DECODER_ADDR_IN	SW ZONE Decoder Address Input	
ACh	EMU_DECODER_OUT	SW ZONE Decoder Output	

Complex bit access types are encoded to fit into small table cells. Table 10-79 shows the codes that are used for access types in this section.

Table 10-79. SSU_CPU1_CFG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		

Table 10-79. SSU_CPU1_CFG_REGS Access Type Codes (continued)

Access Type	Code	Description
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

10.16.3.1 EMU_BOOTDEF_LOW Register (Offset = 28h) [Reset = 00000000h]

EMU_BOOTDEF_LOW is shown in Figure 10-67 and described in Table 10-80.

Return to the Summary Table.

User Emulation Boot Definition Low Register

Figure 10-67. EMU_BOOTDEF_LOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BOOT_DEF3								BOOT_DEF2							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_DEF1								BOOT_DEF0							
R/W-0h								R/W-0h							

Table 10-80. EMU_BOOTDEF_LOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BOOT_DEF3	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
23-16	BOOT_DEF2	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
15-8	BOOT_DEF1	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
7-0	BOOT_DEF0	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. BOOT_DEF0 is the default. Reset type: XRSn

10.16.3.2 EMU_BOOTDEF_HIGH Register (Offset = 2Ch) [Reset = 00000000h]

EMU_BOOTDEF_HIGH is shown in Figure 10-68 and described in Table 10-81.

Return to the Summary Table.

User Emulation Boot Definition High Register

Figure 10-68. EMU_BOOTDEF_HIGH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BOOT_DEF7								BOOT_DEF6							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_DEF5								BOOT_DEF4							
R/W-0h								R/W-0h							

Table 10-81. EMU_BOOTDEF_HIGH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BOOT_DEF7	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
23-16	BOOT_DEF6	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
15-8	BOOT_DEF5	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
7-0	BOOT_DEF4	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn

10.16.3.3 LINK3_CFG Register (Offset = 3Ch) [Reset = 00000002h]

LINK3_CFG is shown in Figure 10-69 and described in Table 10-82.

Return to the Summary Table.

LINK3 Configuration

Figure 10-69. LINK3_CFG Register

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

Table 10-82. LINK3_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK3. When code is decoded to run from LINK3, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.3.4 LINK4_CFG Register (Offset = 40h) [Reset = 00000002h]

LINK4_CFG is shown in Figure 10-70 and described in Table 10-83.

Return to the Summary Table.

LINK4 Configuration

Figure 10-70. LINK4_CFG Register

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

Table 10-83. LINK4_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK4. When code is decoded to run from LINK4, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.3.5 LINK5_CFG Register (Offset = 44h) [Reset = 00000002h]

LINK5_CFG is shown in Figure 10-71 and described in Table 10-84.

Return to the Summary Table.

LINK5 Configuration

Figure 10-71. LINK5_CFG Register

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

Table 10-84. LINK5_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK5. When code is decoded to run from LINK5, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.3.6 LINK6_CFG Register (Offset = 48h) [Reset = 00000002h]

LINK6_CFG is shown in Figure 10-72 and described in Table 10-85.

Return to the Summary Table.

LINK6 Configuration

Figure 10-72. LINK6_CFG Register

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

Table 10-85. LINK6_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK6. When code is decoded to run from LINK6, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.3.7 LINK7_CFG Register (Offset = 4Ch) [Reset = 00000002h]

LINK7_CFG is shown in Figure 10-73 and described in Table 10-86.

Return to the Summary Table.

LINK7 Configuration

Figure 10-73. LINK7_CFG Register

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

Table 10-86. LINK7_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK7. When code is decoded to run from LINK7, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.3.8 LINK8_CFG Register (Offset = 50h) [Reset = 00000002h]

LINK8_CFG is shown in Figure 10-74 and described in Table 10-87.

Return to the Summary Table.

LINK8 Configuration

Figure 10-74. LINK8_CFG Register

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

Table 10-87. LINK8_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK8. When code is decoded to run from LINK8, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.3.9 LINK9_CFG Register (Offset = 54h) [Reset = 00000002h]

LINK9_CFG is shown in Figure 10-75 and described in Table 10-88.

Return to the Summary Table.

LINK9 Configuration

Figure 10-75. LINK9_CFG Register

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

Table 10-88. LINK9_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK9. When code is decoded to run from LINK9, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.3.10 LINK10_CFG Register (Offset = 58h) [Reset = 00000002h]

LINK10_CFG is shown in Figure 10-76 and described in Table 10-89.

Return to the Summary Table.

LINK10 Configuration

Figure 10-76. LINK10_CFG Register

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

Table 10-89. LINK10_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK10. When code is decoded to run from LINK10, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.3.11 LINK11_CFG Register (Offset = 5Ch) [Reset = 00000002h]

LINK11_CFG is shown in Figure 10-77 and described in Table 10-90.

Return to the Summary Table.

LINK11 Configuration

Figure 10-77. LINK11_CFG Register

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

Table 10-90. LINK11_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK11. When code is decoded to run from LINK11, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.3.12 LINK12_CFG Register (Offset = 60h) [Reset = 00000002h]

LINK12_CFG is shown in Figure 10-78 and described in Table 10-91.

Return to the Summary Table.

LINK12 Configuration

Figure 10-78. LINK12_CFG Register

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

Table 10-91. LINK12_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK12. When code is decoded to run from LINK12, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.3.13 LINK13_CFG Register (Offset = 64h) [Reset = 00000002h]

LINK13_CFG is shown in Figure 10-79 and described in Table 10-92.

Return to the Summary Table.

LINK13 Configuration

Figure 10-79. LINK13_CFG Register

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

Table 10-92. LINK13_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK13. When code is decoded to run from LINK13, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.3.14 LINK14_CFG Register (Offset = 68h) [Reset = 00000002h]

LINK14_CFG is shown in Figure 10-80 and described in Table 10-93.

Return to the Summary Table.

LINK14 Configuration

Figure 10-80. LINK14_CFG Register

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

Table 10-93. LINK14_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK14. When code is decoded to run from LINK14, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.3.15 LINK15_CFG Register (Offset = 6Ch) [Reset = 00000002h]

LINK15_CFG is shown in Figure 10-81 and described in Table 10-94.

Return to the Summary Table.

LINK15 Configuration

Figure 10-81. LINK15_CFG Register

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

Table 10-94. LINK15_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK15. When code is decoded to run from LINK15, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.3.16 STACK3_CFG Register (Offset = 7Ch) [Reset = 00000001h]

STACK3_CFG is shown in Figure 10-82 and described in Table 10-95.

Return to the Summary Table.

STACK3 Configuration

Figure 10-82. STACK3_CFG Register

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

Table 10-95. STACK3_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	ZONE	R/W	1h	Defines the ZONE association with STACK3. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : ZONE1 (primary user ZONE) 0x2 : ZONE2 (if available) 0x3 : ZONE3 (if available) Reset type: XRSn

10.16.3.17 STACK4_CFG Register (Offset = 80h) [Reset = 00000001h]

STACK4_CFG is shown in Figure 10-83 and described in Table 10-96.

Return to the Summary Table.

STACK4 Configuration

Figure 10-83. STACK4_CFG Register

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

Table 10-96. STACK4_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	ZONE	R/W	1h	Defines the ZONE association with STACK4. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : ZONE1 (primary user ZONE) 0x2 : ZONE2 (if available) 0x3 : ZONE3 (if available) Reset type: XRSn

10.16.3.18 STACK5_CFG Register (Offset = 84h) [Reset = 00000001h]

STACK5_CFG is shown in Figure 10-84 and described in Table 10-97.

Return to the Summary Table.

STACK5 Configuration

Figure 10-84. STACK5_CFG Register

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

Table 10-97. STACK5_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	ZONE	R/W	1h	Defines the ZONE association with STACK5. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : ZONE1 (primary user ZONE) 0x2 : ZONE2 (if available) 0x3 : ZONE3 (if available) Reset type: XRSn

10.16.3.19 STACK6_CFG Register (Offset = 88h) [Reset = 00000001h]

STACK6_CFG is shown in Figure 10-85 and described in Table 10-98.

Return to the Summary Table.

STACK6 Configuration

Figure 10-85. STACK6_CFG Register

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

Table 10-98. STACK6_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	ZONE	R/W	1h	Defines the ZONE association with STACK6. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : ZONE1 (primary user ZONE) 0x2 : ZONE2 (if available) 0x3 : ZONE3 (if available) Reset type: XRSn

10.16.3.20 STACK7_CFG Register (Offset = 8Ch) [Reset = 00000001h]

STACK7_CFG is shown in Figure 10-86 and described in Table 10-99.

Return to the Summary Table.

STACK7 Configuration

Figure 10-86. STACK7_CFG Register

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

Table 10-99. STACK7_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	ZONE	R/W	1h	Defines the ZONE association with STACK7. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : ZONE1 (primary user ZONE) 0x2 : ZONE2 (if available) 0x3 : ZONE3 (if available) Reset type: XRSn

10.16.3.21 RAMOPENSTAT Register (Offset = 90h) [Reset = 00000000h]

RAMOPENSTAT is shown in Figure 10-87 and described in Table 10-100.

Return to the Summary Table.

RAMOPEN Feature Status Register

Figure 10-87. RAMOPENSTAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						LINK1_RAMOPENS	RESERVED
R-0-0h						R-0h	R-0h

Table 10-100. RAMOPENSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	LINK1_RAMOPENS	R	0h	Reflects the status of the RAMOPEN feature for LINK1. 0 : RAMOPEN is not active. RAMs are assigned per the AP registers. 1 : RAMOPEN is active. Specified RAMs are assigned to LINK1. Reset type: XRSn
0	RESERVED	R	0h	Reserved

10.16.3.22 RAMOPENFRC Register (Offset = 94h) [Reset = 00000000h]

RAMOPENFRC is shown in Figure 10-88 and described in Table 10-101.

Return to the Summary Table.

RAMOPEN Feature Force Register

Figure 10-88. RAMOPENFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED			WIPE_ON_LINK1_CLR	RESERVED			SET_LINK1
R-0-0h			R/W1S-0h	R-0-0h			R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED			RESERVED	RESERVED			RESERVED
R-0-0h			R/W1S-0h	R-0-0h			R-0/W1S-0h

Table 10-101. RAMOPENFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-13	RESERVED	R-0	0h	Reserved
12	WIPE_ON_LINK1_CLR	R/W1S	0h	Defines if the HW goes through a RAMINIT when it closes RAMOPEN for LINK1. This bit is cleared after RAMOPEN is closed via RAMOPENCLR.CLEAR_LINK1. Reset type: XRSn
11-9	RESERVED	R-0	0h	Reserved
8	SET_LINK1	R-0/W1S	0h	Writing a '1' to this bit requests the 'RAMOPEN' mode for LINK1. When this bit is written with '1', hardware wipes out the RAM content using the RAMINIT feature and then sets the RAMOPENSTAT.RAMOPEN bit after completion of RAMINIT for all RAMOPEN-able RAMs. Writes of '0' are ignored. Reset type: XRSn
7-5	RESERVED	R-0	0h	Reserved
4	RESERVED	R/W1S	0h	Reserved
3-1	RESERVED	R-0	0h	Reserved
0	RESERVED	R-0/W1S	0h	Reserved

10.16.3.23 RAMOPENCLR Register (Offset = 98h) [Reset = 00000000h]

RAMOPENCLR is shown in Figure 10-89 and described in Table 10-102.

Return to the Summary Table.

RAMOPEN Feature Clear Register

Figure 10-89. RAMOPENCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						CLEAR_LINK1	RESERVED
R-0-0h						R-0/W1S-0h	R-0/W1S-0h

Table 10-102. RAMOPENCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-2	RESERVED	R-0	0h	Reserved
1	CLEAR_LINK1	R-0/W1S	0h	Writing a '1' to this bit clears the RAMOPENSTAT.LINK1_RAMOPENS bit, causing LINK1 to come out of 'RAMOPEN' mode. If RAMOPENFRC.WIPE_ON_LINK1_CLR is set, hardware wipes out the LINK1 RAMOPEN-able RAM content using the RAMINIT feature prior to clearing the RAMOPENSTAT.LINK1_RAMOPENS bit. RAMOPENSET.WIPE_ON_LINK1_CLR is also cleared when this bit is written with a '1'. Writes of '0' are ignored. Reset type: XRSn
0	RESERVED	R-0/W1S	0h	Reserved

10.16.3.24 DECODER_ADDR_IN Register (Offset = A0h) [Reset = 00000000h]

DECODER_ADDR_IN is shown in Figure 10-90 and described in Table 10-103.

Return to the Summary Table.

SW ZONE Decoder Address Input

Figure 10-90. DECODER_ADDR_IN Register

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

Table 10-103. DECODER_ADDR_IN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ADDR_H	R/W	0h	See ADDR_L. Reset type: XRSn
15-12	ADDR_L	R/W	0h	The 4KB address written to this register is evaluated against the user AP regions to determine the LINK, STACK, and ZONE of the address. The output is provided in DECODER_OUT. The Access Protection region number responding to the request is also provided. Reset type: XRSn
11-0	RESERVED	R	0h	Reserved

10.16.3.25 DECODER_OUT Register (Offset = A4h) [Reset = 80000000h]

DECODER_OUT is shown in Figure 10-91 and described in Table 10-104.

Return to the Summary Table.

SW ZONE Decoder Output

Figure 10-91. DECODER_OUT Register

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

Table 10-104. DECODER_OUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INVALID	R	1h	If either no AP region or multiple AP regions respond to the address query requested in DECODER_ADDR_IN, this bit will be set. If set, the contents in this register are not valid. Reset type: XRSn
30-23	RESERVED	R-0	0h	Reserved
22-16	APR	R	0h	See LINK. In SSUMODE1, this field will always read as a zero. Reset type: XRSn
15-10	RESERVED	R-0	0h	Reserved
9-8	ZONE	R	0h	See LINK. Reset type: XRSn
7	RESERVED	R-0	0h	Reserved
6-4	STACK	R	0h	See LINK. Reset type: XRSn
3-0	LINK	R	0h	The output from DECODER_ADDR_IN. Reset type: XRSn

10.16.3.26 EMU_DECODER_ADDR_IN Register (Offset = A8h) [Reset = 00000000h]

EMU_DECODER_ADDR_IN is shown in Figure 10-92 and described in Table 10-105.

Return to the Summary Table.

SW ZONE Decoder Address Input

Figure 10-92. EMU_DECODER_ADDR_IN Register

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

Table 10-105. EMU_DECODER_ADDR_IN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ADDR_H	R/W	0h	See ADDR_L. Reset type: XRSn
15-12	ADDR_L	R/W	0h	The 4KB address written to this register is evaluated against the user AP regions to determine the LINK, STACK, and ZONE of the address. The output is provided in DECODER_OUT. The Access Protection region number responding to the request is also provided. Debugger accesses should use this register and its corresponding output register and reserve the DECODER_ADDR_IN/OUT registers for runtime use. Reset type: XRSn
11-0	RESERVED	R	0h	Reserved

10.16.3.27 EMU_DECODER_OUT Register (Offset = ACh) [Reset = 80000000h]

EMU_DECODER_OUT is shown in Figure 10-93 and described in Table 10-106.

Return to the Summary Table.

SW ZONE Decoder Output

Figure 10-93. EMU_DECODER_OUT Register

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

Table 10-106. EMU_DECODER_OUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INVALID	R	1h	If either no AP region or multiple AP regions respond to the address query requested in EMU_DECODER_ADDR_IN, this bit will be set. If set, the contents in this register are not valid. Reset type: XRSn
30-23	RESERVED	R-0	0h	Reserved
22-16	APR	R	0h	See LINK. In SSUMODE1, this field will always read as a zero. Reset type: XRSn
15-10	RESERVED	R-0	0h	Reserved
9-8	ZONE	R	0h	See LINK. Reset type: XRSn
7	RESERVED	R-0	0h	Reserved
6-4	STACK	R	0h	See LINK. Reset type: XRSn
3-0	LINK	R	0h	The output from EMU_DECODER_ADDR_IN. Reset type: XRSn

10.16.4 SSU_CPU2_CFG_REGS Registers

Table 10-107 lists the memory-mapped registers for the SSU_CPU2_CFG_REGS registers. All register offset addresses not listed in Table 10-107 should be considered as reserved locations and the register contents should not be modified.

Table 10-107. SSU_CPU2_CFG_REGS Registers

Offset	Acronym	Register Name	Protection
0h	RST_VECT	CPU Reset Vector	
4h	RST_LINK	CPU Reset LINK	
8h	CPU_RST_CTRL	CPU Reset Control	
10h	DEF_NMI_VECT	Default CPU NMI Vector	
14h	DEF_NMI_LINK	Default CPU NMI LINK	
28h	EMU_BOOTDEF_LOW	User Emulation Boot Definition Low Register	
2Ch	EMU_BOOTDEF_HIGH	User Emulation Boot Definition High Register	
3Ch	LINK3_CFG	LINK3 Configuration	
40h	LINK4_CFG	LINK4 Configuration	
44h	LINK5_CFG	LINK5 Configuration	
48h	LINK6_CFG	LINK6 Configuration	
4Ch	LINK7_CFG	LINK7 Configuration	
50h	LINK8_CFG	LINK8 Configuration	
54h	LINK9_CFG	LINK9 Configuration	
58h	LINK10_CFG	LINK10 Configuration	
5Ch	LINK11_CFG	LINK11 Configuration	
60h	LINK12_CFG	LINK12 Configuration	
64h	LINK13_CFG	LINK13 Configuration	
68h	LINK14_CFG	LINK14 Configuration	
6Ch	LINK15_CFG	LINK15 Configuration	
7Ch	STACK3_CFG	STACK3 Configuration	
80h	STACK4_CFG	STACK4 Configuration	
84h	STACK5_CFG	STACK5 Configuration	
88h	STACK6_CFG	STACK6 Configuration	
8Ch	STACK7_CFG	STACK7 Configuration	
90h	RAMOPENSTAT	RAMOPEN Feature Status Register	
94h	RAMOPENFRC	RAMOPEN Feature Force Register	
98h	RAMOPENCLR	RAMOPEN Feature Clear Register	
A0h	DECODER_ADDR_IN	SW ZONE Decoder Address Input	
A4h	DECODER_OUT	SW ZONE Decoder Output	
A8h	EMU_DECODER_ADDR_IN	SW ZONE Decoder Address Input	
ACh	EMU_DECODER_OUT	SW ZONE Decoder Output	

Complex bit access types are encoded to fit into small table cells. Table 10-108 shows the codes that are used for access types in this section.

Table 10-108. SSU_CPU2_CFG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s

Table 10-108. SSU_CPU2_CFG_REGS Access Type Codes (continued)

Access Type	Code	Description
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

10.16.4.1 RST_VECT Register (Offset = 0h) [Reset = 00000000h]

RST_VECT is shown in Figure 10-94 and described in Table 10-109.

Return to the Summary Table.

CPU Reset Vector

Figure 10-94. RST_VECT Register

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

Table 10-109. RST_VECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R/W	0h	Defines the address to which the CPU will boot. Loaded by CPU1.LINK2 application code prior to releasing this CPU's reset. Reset type: XRSn

10.16.4.2 RST_LINK Register (Offset = 4h) [Reset = 00000000h]

RST_LINK is shown in Figure 10-95 and described in Table 10-110.

Return to the Summary Table.

CPU Reset LINK

Figure 10-95. RST_LINK Register

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

Table 10-110. RST_LINK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	RESERVED	R	0h	Reserved
3-0	LINK	R/W	0h	Defines the LINK to which the CPU will boot. Loaded by CPU1.LINK2 application code prior to releasing this CPU's reset. Reset type: XRSn

10.16.4.3 CPU_RST_CTRL Register (Offset = 8h) [Reset = 000000C9h]

CPU_RST_CTRL is shown in Figure 10-96 and described in Table 10-111.

Return to the Summary Table.

CPU Reset Control

Figure 10-96. CPU_RST_CTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SW_SYSRSN							
R-0h								R/W-C9h							

Table 10-111. CPU_RST_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	RESERVED	R	0h	Reserved
7-0	SW_SYSRSN	R/W	C9h	While this bit is low, the CPU is held in reset. Once set high, if also released by the HSM (if present on the device), this CPU is released from reset. 0xC9 : CPU in reset 0x36 : CPU reset is released (if no HSM) or determined by HSM input Others : CPU in reset Reset type: CPU1.SYSRSn

10.16.4.4 DEF_NMI_VECT Register (Offset = 10h) [Reset = 00000000h]

DEF_NMI_VECT is shown in Figure 10-97 and described in Table 10-112.

Return to the Summary Table.

Default CPU NMI Vector

Figure 10-97. DEF_NMI_VECT Register

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

Table 10-112. DEF_NMI_VECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R/W	0h	Defines the reset address of the NMI_VECT register in the PIPE. Loaded by CPU1.LINK2 application code prior to releasing this CPU's reset. Reset type: XRSn

10.16.4.5 DEF_NMI_LINK Register (Offset = 14h) [Reset = 00000000h]

DEF_NMI_LINK is shown in Figure 10-98 and described in Table 10-113.

Return to the Summary Table.

Default CPU NMI LINK

Figure 10-98. DEF_NMI_LINK Register

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

Table 10-113. DEF_NMI_LINK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	RESERVED	R	0h	Reserved
3-0	LINK	R/W	0h	Defines the reset LINK of the NMI_LINK register in the PIPE. Loaded by CPU1.LINK2 application code prior to releasing this CPU's reset. Reset type: XRSn

10.16.4.6 EMU_BOOTDEF_LOW Register (Offset = 28h) [Reset = 00000000h]

EMU_BOOTDEF_LOW is shown in Figure 10-99 and described in Table 10-114.

Return to the Summary Table.

User Emulation Boot Definition Low Register

Figure 10-99. EMU_BOOTDEF_LOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BOOT_DEF3								BOOT_DEF2							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_DEF1								BOOT_DEF0							
R/W-0h								R/W-0h							

Table 10-114. EMU_BOOTDEF_LOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BOOT_DEF3	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
23-16	BOOT_DEF2	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
15-8	BOOT_DEF1	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
7-0	BOOT_DEF0	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. BOOT_DEF0 is the default. Reset type: XRSn

10.16.4.7 EMU_BOOTDEF_HIGH Register (Offset = 2Ch) [Reset = 00000000h]

EMU_BOOTDEF_HIGH is shown in Figure 10-100 and described in Table 10-115.

Return to the Summary Table.

User Emulation Boot Definition High Register

Figure 10-100. EMU_BOOTDEF_HIGH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BOOT_DEF7								BOOT_DEF6							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_DEF5								BOOT_DEF4							
R/W-0h								R/W-0h							

Table 10-115. EMU_BOOTDEF_HIGH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BOOT_DEF7	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
23-16	BOOT_DEF6	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
15-8	BOOT_DEF5	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
7-0	BOOT_DEF4	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn

10.16.4.8 LINK3_CFG Register (Offset = 3Ch) [Reset = 00000002h]

LINK3_CFG is shown in Figure 10-101 and described in Table 10-116.

Return to the Summary Table.

LINK3 Configuration

Figure 10-101. LINK3_CFG Register

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

Table 10-116. LINK3_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK3. When code is decoded to run from LINK3, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.4.9 LINK4_CFG Register (Offset = 40h) [Reset = 00000002h]

LINK4_CFG is shown in Figure 10-102 and described in Table 10-117.

Return to the Summary Table.

LINK4 Configuration

Figure 10-102. LINK4_CFG Register

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

Table 10-117. LINK4_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK4. When code is decoded to run from LINK4, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.4.10 LINK5_CFG Register (Offset = 44h) [Reset = 00000002h]

LINK5_CFG is shown in Figure 10-103 and described in Table 10-118.

Return to the Summary Table.

LINK5 Configuration

Figure 10-103. LINK5_CFG Register

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

Table 10-118. LINK5_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK5. When code is decoded to run from LINK5, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.4.11 LINK6_CFG Register (Offset = 48h) [Reset = 00000002h]

LINK6_CFG is shown in Figure 10-104 and described in Table 10-119.

Return to the Summary Table.

LINK6 Configuration

Figure 10-104. LINK6_CFG Register

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

Table 10-119. LINK6_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK6. When code is decoded to run from LINK6, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.4.12 LINK7_CFG Register (Offset = 4Ch) [Reset = 00000002h]

LINK7_CFG is shown in Figure 10-105 and described in Table 10-120.

Return to the Summary Table.

LINK7 Configuration

Figure 10-105. LINK7_CFG Register

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

Table 10-120. LINK7_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK7. When code is decoded to run from LINK7, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.4.13 LINK8_CFG Register (Offset = 50h) [Reset = 00000002h]

LINK8_CFG is shown in Figure 10-106 and described in Table 10-121.

Return to the Summary Table.

LINK8 Configuration

Figure 10-106. LINK8_CFG Register

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

Table 10-121. LINK8_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK8. When code is decoded to run from LINK8, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.4.14 LINK9_CFG Register (Offset = 54h) [Reset = 00000002h]

LINK9_CFG is shown in Figure 10-107 and described in Table 10-122.

Return to the Summary Table.

LINK9 Configuration

Figure 10-107. LINK9_CFG Register

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

Table 10-122. LINK9_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK9. When code is decoded to run from LINK9, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.4.15 LINK10_CFG Register (Offset = 58h) [Reset = 00000002h]

LINK10_CFG is shown in Figure 10-108 and described in Table 10-123.

Return to the Summary Table.

LINK10 Configuration

Figure 10-108. LINK10_CFG Register

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

Table 10-123. LINK10_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK10. When code is decoded to run from LINK10, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.4.16 LINK11_CFG Register (Offset = 5Ch) [Reset = 00000002h]

LINK11_CFG is shown in Figure 10-109 and described in Table 10-124.

Return to the Summary Table.

LINK11 Configuration

Figure 10-109. LINK11_CFG Register

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

Table 10-124. LINK11_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK11. When code is decoded to run from LINK11, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.4.17 LINK12_CFG Register (Offset = 60h) [Reset = 00000002h]

LINK12_CFG is shown in Figure 10-110 and described in Table 10-125.

Return to the Summary Table.

LINK12 Configuration

Figure 10-110. LINK12_CFG Register

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

Table 10-125. LINK12_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK12. When code is decoded to run from LINK12, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.4.18 LINK13_CFG Register (Offset = 64h) [Reset = 00000002h]

LINK13_CFG is shown in Figure 10-111 and described in Table 10-126.

Return to the Summary Table.

LINK13 Configuration

Figure 10-111. LINK13_CFG Register

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

Table 10-126. LINK13_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK13. When code is decoded to run from LINK13, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.4.19 LINK14_CFG Register (Offset = 68h) [Reset = 00000002h]

LINK14_CFG is shown in Figure 10-112 and described in Table 10-127.

Return to the Summary Table.

LINK14 Configuration

Figure 10-112. LINK14_CFG Register

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

Table 10-127. LINK14_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK14. When code is decoded to run from LINK14, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.4.20 LINK15_CFG Register (Offset = 6Ch) [Reset = 00000002h]

LINK15_CFG is shown in Figure 10-113 and described in Table 10-128.

Return to the Summary Table.

LINK15 Configuration

Figure 10-113. LINK15_CFG Register

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

Table 10-128. LINK15_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK15. When code is decoded to run from LINK15, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.4.21 STACK3_CFG Register (Offset = 7Ch) [Reset = 00000001h]

STACK3_CFG is shown in Figure 10-114 and described in Table 10-129.

Return to the Summary Table.

STACK3 Configuration

Figure 10-114. STACK3_CFG Register

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

Table 10-129. STACK3_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	ZONE	R/W	1h	Defines the ZONE association with STACK3. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : ZONE1 (primary user ZONE) 0x2 : ZONE2 (if available) 0x3 : ZONE3 (if available) Reset type: XRSn

10.16.4.22 STACK4_CFG Register (Offset = 80h) [Reset = 00000001h]

STACK4_CFG is shown in Figure 10-115 and described in Table 10-130.

Return to the Summary Table.

STACK4 Configuration

Figure 10-115. STACK4_CFG Register

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

Table 10-130. STACK4_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	ZONE	R/W	1h	Defines the ZONE association with STACK4. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : ZONE1 (primary user ZONE) 0x2 : ZONE2 (if available) 0x3 : ZONE3 (if available) Reset type: XRSn

10.16.4.23 STACK5_CFG Register (Offset = 84h) [Reset = 00000001h]

STACK5_CFG is shown in Figure 10-116 and described in Table 10-131.

Return to the Summary Table.

STACK5 Configuration

Figure 10-116. STACK5_CFG Register

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

Table 10-131. STACK5_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	ZONE	R/W	1h	Defines the ZONE association with STACK5. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : ZONE1 (primary user ZONE) 0x2 : ZONE2 (if available) 0x3 : ZONE3 (if available) Reset type: XRSn

10.16.4.24 STACK6_CFG Register (Offset = 88h) [Reset = 00000001h]

STACK6_CFG is shown in Figure 10-117 and described in Table 10-132.

Return to the Summary Table.

STACK6 Configuration

Figure 10-117. STACK6_CFG Register

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

Table 10-132. STACK6_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	ZONE	R/W	1h	Defines the ZONE association with STACK6. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : ZONE1 (primary user ZONE) 0x2 : ZONE2 (if available) 0x3 : ZONE3 (if available) Reset type: XRSn

10.16.4.25 STACK7_CFG Register (Offset = 8Ch) [Reset = 00000001h]

STACK7_CFG is shown in Figure 10-118 and described in Table 10-133.

Return to the Summary Table.

STACK7 Configuration

Figure 10-118. STACK7_CFG Register

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

Table 10-133. STACK7_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	ZONE	R/W	1h	Defines the ZONE association with STACK7. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : ZONE1 (primary user ZONE) 0x2 : ZONE2 (if available) 0x3 : ZONE3 (if available) Reset type: XRSn

10.16.4.26 RAMOPENSTAT Register (Offset = 90h) [Reset = 00000000h]

RAMOPENSTAT is shown in Figure 10-119 and described in Table 10-134.

Return to the Summary Table.

RAMOPEN Feature Status Register

Figure 10-119. RAMOPENSTAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						LINK1_RAMOPENS	RESERVED
R-0-0h						R-0h	R-0h

Table 10-134. RAMOPENSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	LINK1_RAMOPENS	R	0h	Reflects the status of the RAMOPEN feature for LINK1. 0 : RAMOPEN is not active. RAMs are assigned per the AP registers. 1 : RAMOPEN is active. Specified RAMs are assigned to LINK1. Reset type: XRSn
0	RESERVED	R	0h	Reserved

10.16.4.27 RAMOPENFRC Register (Offset = 94h) [Reset = 00000000h]

RAMOPENFRC is shown in Figure 10-120 and described in Table 10-135.

Return to the Summary Table.

RAMOPEN Feature Force Register

Figure 10-120. RAMOPENFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED			WIPE_ON_LINK1_CLR	RESERVED			SET_LINK1
R-0-0h			R/W1S-0h	R-0-0h			R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED			RESERVED	RESERVED			RESERVED
R-0-0h			R/W1S-0h	R-0-0h			R-0/W1S-0h

Table 10-135. RAMOPENFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-13	RESERVED	R-0	0h	Reserved
12	WIPE_ON_LINK1_CLR	R/W1S	0h	Defines if the HW goes through a RAMINIT when it closes RAMOPEN for LINK1. This bit is cleared after RAMOPEN is closed via RAMOPENCLR.CLEAR_LINK1. Reset type: XRSn
11-9	RESERVED	R-0	0h	Reserved
8	SET_LINK1	R-0/W1S	0h	Writing a '1' to this bit requests the 'RAMOPEN' mode for LINK1. When this bit is written with '1', hardware wipes out the RAM content using the RAMINIT feature and then sets the RAMOPENSTAT.RAMOPEN bit after completion of RAMINIT for all RAMOPEN-able RAMs. Writes of '0' are ignored. Reset type: XRSn
7-5	RESERVED	R-0	0h	Reserved
4	RESERVED	R/W1S	0h	Reserved
3-1	RESERVED	R-0	0h	Reserved
0	RESERVED	R-0/W1S	0h	Reserved

10.16.4.28 RAMOPENCLR Register (Offset = 98h) [Reset = 00000000h]

RAMOPENCLR is shown in Figure 10-121 and described in Table 10-136.

Return to the Summary Table.

RAMOPEN Feature Clear Register

Figure 10-121. RAMOPENCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						CLEAR_LINK1	RESERVED
R-0-0h						R-0/W1S-0h	R-0/W1S-0h

Table 10-136. RAMOPENCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-2	RESERVED	R-0	0h	Reserved
1	CLEAR_LINK1	R-0/W1S	0h	Writing a '1' to this bit clears the RAMOPENSTAT.LINK1_RAMOPENS bit, causing LINK1 to come out of 'RAMOPEN' mode. If RAMOPENFRC.WIPE_ON_LINK1_CLR is set, hardware wipes out the LINK1 RAMOPEN-able RAM content using the RAMINIT feature prior to clearing the RAMOPENSTAT.LINK1_RAMOPENS bit. RAMOPENSET.WIPE_ON_LINK1_CLR is also cleared when this bit is written with a '1'. Writes of '0' are ignored. Reset type: XRSn
0	RESERVED	R-0/W1S	0h	Reserved

10.16.4.29 DECODER_ADDR_IN Register (Offset = A0h) [Reset = 00000000h]

DECODER_ADDR_IN is shown in Figure 10-122 and described in Table 10-137.

Return to the Summary Table.

SW ZONE Decoder Address Input

Figure 10-122. DECODER_ADDR_IN Register

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

Table 10-137. DECODER_ADDR_IN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ADDR_H	R/W	0h	See ADDR_L. Reset type: XRSn
15-12	ADDR_L	R/W	0h	The 4KB address written to this register is evaluated against the user AP regions to determine the LINK, STACK, and ZONE of the address. The output is provided in DECODER_OUT. The Access Protection region number responding to the request is also provided. Reset type: XRSn
11-0	RESERVED	R	0h	Reserved

10.16.4.30 DECODER_OUT Register (Offset = A4h) [Reset = 80000000h]

DECODER_OUT is shown in Figure 10-123 and described in Table 10-138.

Return to the Summary Table.

SW ZONE Decoder Output

Figure 10-123. DECODER_OUT Register

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

Table 10-138. DECODER_OUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INVALID	R	1h	If either no AP region or multiple AP regions respond to the address query requested in DECODER_ADDR_IN, this bit will be set. If set, the contents in this register are not valid. Reset type: XRSn
30-23	RESERVED	R-0	0h	Reserved
22-16	APR	R	0h	See LINK. In SSUMODE1, this field will always read as a zero. Reset type: XRSn
15-10	RESERVED	R-0	0h	Reserved
9-8	ZONE	R	0h	See LINK. Reset type: XRSn
7	RESERVED	R-0	0h	Reserved
6-4	STACK	R	0h	See LINK. Reset type: XRSn
3-0	LINK	R	0h	The output from DECODER_ADDR_IN. Reset type: XRSn

10.16.4.31 EMU_DECODER_ADDR_IN Register (Offset = A8h) [Reset = 00000000h]

EMU_DECODER_ADDR_IN is shown in Figure 10-124 and described in Table 10-139.

Return to the Summary Table.

SW ZONE Decoder Address Input

Figure 10-124. EMU_DECODER_ADDR_IN Register

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

Table 10-139. EMU_DECODER_ADDR_IN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ADDR_H	R/W	0h	See ADDR_L. Reset type: XRSn
15-12	ADDR_L	R/W	0h	The 4KB address written to this register is evaluated against the user AP regions to determine the LINK, STACK, and ZONE of the address. The output is provided in DECODER_OUT. The Access Protection region number responding to the request is also provided. Debugger accesses should use this register and its corresponding output register and reserve the DECODER_ADDR_IN/OUT registers for runtime use. Reset type: XRSn
11-0	RESERVED	R	0h	Reserved

10.16.4.32 EMU_DECODER_OUT Register (Offset = ACh) [Reset = 80000000h]

EMU_DECODER_OUT is shown in Figure 10-125 and described in Table 10-140.

Return to the Summary Table.

SW ZONE Decoder Output

Figure 10-125. EMU_DECODER_OUT Register

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

Table 10-140. EMU_DECODER_OUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INVALID	R	1h	If either no AP region or multiple AP regions respond to the address query requested in EMU_DECODER_ADDR_IN, this bit will be set. If set, the contents in this register are not valid. Reset type: XRSn
30-23	RESERVED	R-0	0h	Reserved
22-16	APR	R	0h	See LINK. In SSUMODE1, this field will always read as a zero. Reset type: XRSn
15-10	RESERVED	R-0	0h	Reserved
9-8	ZONE	R	0h	See LINK. Reset type: XRSn
7	RESERVED	R-0	0h	Reserved
6-4	STACK	R	0h	See LINK. Reset type: XRSn
3-0	LINK	R	0h	The output from EMU_DECODER_ADDR_IN. Reset type: XRSn

10.16.5 SSU_CPU3_CFG_REGS Registers

Table 10-141 lists the memory-mapped registers for the SSU_CPU3_CFG_REGS registers. All register offset addresses not listed in Table 10-141 should be considered as reserved locations and the register contents should not be modified.

Table 10-141. SSU_CPU3_CFG_REGS Registers

Offset	Acronym	Register Name	Protection
0h	RST_VECT	CPU Reset Vector	
4h	RST_LINK	CPU Reset LINK	
8h	CPU_RST_CTRL	CPU Reset Control	
10h	DEF_NMI_VECT	Default CPU NMI Vector	
14h	DEF_NMI_LINK	Default CPU NMI LINK	
28h	EMU_BOOTDEF_LOW	User Emulation Boot Definition Low Register	
2Ch	EMU_BOOTDEF_HIGH	User Emulation Boot Definition High Register	
3Ch	LINK3_CFG	LINK3 Configuration	
40h	LINK4_CFG	LINK4 Configuration	
44h	LINK5_CFG	LINK5 Configuration	
48h	LINK6_CFG	LINK6 Configuration	
4Ch	LINK7_CFG	LINK7 Configuration	
50h	LINK8_CFG	LINK8 Configuration	
54h	LINK9_CFG	LINK9 Configuration	
58h	LINK10_CFG	LINK10 Configuration	
5Ch	LINK11_CFG	LINK11 Configuration	
60h	LINK12_CFG	LINK12 Configuration	
64h	LINK13_CFG	LINK13 Configuration	
68h	LINK14_CFG	LINK14 Configuration	
6Ch	LINK15_CFG	LINK15 Configuration	
7Ch	STACK3_CFG	STACK3 Configuration	
80h	STACK4_CFG	STACK4 Configuration	
84h	STACK5_CFG	STACK5 Configuration	
88h	STACK6_CFG	STACK6 Configuration	
8Ch	STACK7_CFG	STACK7 Configuration	
90h	RAMOPENSTAT	RAMOPEN Feature Status Register	
94h	RAMOPENFRC	RAMOPEN Feature Force Register	
98h	RAMOPENCLR	RAMOPEN Feature Clear Register	
A0h	DECODER_ADDR_IN	SW ZONE Decoder Address Input	
A4h	DECODER_OUT	SW ZONE Decoder Output	
A8h	EMU_DECODER_ADDR_IN	SW ZONE Decoder Address Input	
ACh	EMU_DECODER_OUT	SW ZONE Decoder Output	

Complex bit access types are encoded to fit into small table cells. Table 10-142 shows the codes that are used for access types in this section.

Table 10-142. SSU_CPU3_CFG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s

Table 10-142. SSU_CPU3_CFG_REGS Access Type Codes (continued)

Access Type	Code	Description
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

10.16.5.1 RST_VECT Register (Offset = 0h) [Reset = 00000000h]

RST_VECT is shown in Figure 10-126 and described in Table 10-143.

Return to the Summary Table.

CPU Reset Vector

Figure 10-126. RST_VECT Register

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

Table 10-143. RST_VECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R/W	0h	Defines the address to which the CPU will boot. Loaded by CPU1.LINK2 application code prior to releasing this CPU's reset. Reset type: XRSn

10.16.5.2 RST_LINK Register (Offset = 4h) [Reset = 00000000h]

RST_LINK is shown in Figure 10-127 and described in Table 10-144.

Return to the Summary Table.

CPU Reset LINK

Figure 10-127. RST_LINK Register

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

Table 10-144. RST_LINK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	RESERVED	R	0h	Reserved
3-0	LINK	R/W	0h	Defines the LINK to which the CPU will boot. Loaded by CPU1.LINK2 application code prior to releasing this CPU's reset. Reset type: XRSn

10.16.5.3 CPU_RST_CTRL Register (Offset = 8h) [Reset = 000000C9h]

CPU_RST_CTRL is shown in Figure 10-128 and described in Table 10-145.

Return to the Summary Table.

CPU Reset Control

Figure 10-128. CPU_RST_CTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SW_SYSRSN							
R-0h								R/W-C9h							

Table 10-145. CPU_RST_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	RESERVED	R	0h	Reserved
7-0	SW_SYSRSN	R/W	C9h	While this bit is low, the CPU is held in reset. Once set high, if also released by the HSM (if present on the device), this CPU is released from reset. 0xC9 : CPU in reset 0x36 : CPU reset is released (if no HSM) or determined by HSM input Others : CPU in reset Reset type: CPU1.SYSRSn

10.16.5.4 DEF_NMI_VECT Register (Offset = 10h) [Reset = 00000000h]

DEF_NMI_VECT is shown in Figure 10-129 and described in Table 10-146.

Return to the Summary Table.

Default CPU NMI Vector

Figure 10-129. DEF_NMI_VECT Register

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

Table 10-146. DEF_NMI_VECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R/W	0h	Defines the reset address of the NMI_VECT register in the PIPE. Loaded by CPU1.LINK2 application code prior to releasing this CPU's reset. Reset type: XRSn

10.16.5.5 DEF_NMI_LINK Register (Offset = 14h) [Reset = 00000000h]

DEF_NMI_LINK is shown in Figure 10-130 and described in Table 10-147.

Return to the Summary Table.

Default CPU NMI LINK

Figure 10-130. DEF_NMI_LINK Register

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

Table 10-147. DEF_NMI_LINK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	RESERVED	R	0h	Reserved
3-0	LINK	R/W	0h	Defines the reset LINK of the NMI_LINK register in the PIPE. Loaded by CPU1.LINK2 application code prior to releasing this CPU's reset. Reset type: XRSn

10.16.5.6 EMU_BOOTDEF_LOW Register (Offset = 28h) [Reset = 00000000h]

EMU_BOOTDEF_LOW is shown in Figure 10-131 and described in Table 10-148.

Return to the Summary Table.

User Emulation Boot Definition Low Register

Figure 10-131. EMU_BOOTDEF_LOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BOOT_DEF3								BOOT_DEF2							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_DEF1								BOOT_DEF0							
R/W-0h								R/W-0h							

Table 10-148. EMU_BOOTDEF_LOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BOOT_DEF3	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
23-16	BOOT_DEF2	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
15-8	BOOT_DEF1	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
7-0	BOOT_DEF0	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. BOOT_DEF0 is the default. Reset type: XRSn

10.16.5.7 EMU_BOOTDEF_HIGH Register (Offset = 2Ch) [Reset = 00000000h]

EMU_BOOTDEF_HIGH is shown in Figure 10-132 and described in Table 10-149.

Return to the Summary Table.

User Emulation Boot Definition High Register

Figure 10-132. EMU_BOOTDEF_HIGH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BOOT_DEF7								BOOT_DEF6							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_DEF5								BOOT_DEF4							
R/W-0h								R/W-0h							

Table 10-149. EMU_BOOTDEF_HIGH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BOOT_DEF7	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
23-16	BOOT_DEF6	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
15-8	BOOT_DEF5	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn
7-0	BOOT_DEF4	R/W	0h	Defines the emulation boot method when chosen by the user configuration/pins. Reset type: XRSn

10.16.5.8 LINK3_CFG Register (Offset = 3Ch) [Reset = 00000002h]

LINK3_CFG is shown in Figure 10-133 and described in Table 10-150.

Return to the Summary Table.

LINK3 Configuration

Figure 10-133. LINK3_CFG Register

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

Table 10-150. LINK3_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK3. When code is decoded to run from LINK3, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.5.9 LINK4_CFG Register (Offset = 40h) [Reset = 00000002h]

LINK4_CFG is shown in Figure 10-134 and described in Table 10-151.

Return to the Summary Table.

LINK4 Configuration

Figure 10-134. LINK4_CFG Register

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

Table 10-151. LINK4_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK4. When code is decoded to run from LINK4, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.5.10 LINK5_CFG Register (Offset = 44h) [Reset = 00000002h]

LINK5_CFG is shown in Figure 10-135 and described in Table 10-152.

Return to the Summary Table.

LINK5 Configuration

Figure 10-135. LINK5_CFG Register

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

Table 10-152. LINK5_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK5. When code is decoded to run from LINK5, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.5.11 LINK6_CFG Register (Offset = 48h) [Reset = 00000002h]

LINK6_CFG is shown in Figure 10-136 and described in Table 10-153.

Return to the Summary Table.

LINK6 Configuration

Figure 10-136. LINK6_CFG Register

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

Table 10-153. LINK6_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK6. When code is decoded to run from LINK6, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.5.12 LINK7_CFG Register (Offset = 4Ch) [Reset = 00000002h]

LINK7_CFG is shown in Figure 10-137 and described in Table 10-154.

Return to the Summary Table.

LINK7 Configuration

Figure 10-137. LINK7_CFG Register

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

Table 10-154. LINK7_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK7. When code is decoded to run from LINK7, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.5.13 LINK8_CFG Register (Offset = 50h) [Reset = 00000002h]

LINK8_CFG is shown in Figure 10-138 and described in Table 10-155.

Return to the Summary Table.

LINK8 Configuration

Figure 10-138. LINK8_CFG Register

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

Table 10-155. LINK8_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK8. When code is decoded to run from LINK8, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.5.14 LINK9_CFG Register (Offset = 54h) [Reset = 00000002h]

LINK9_CFG is shown in Figure 10-139 and described in Table 10-156.

Return to the Summary Table.

LINK9 Configuration

Figure 10-139. LINK9_CFG Register

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

Table 10-156. LINK9_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK9. When code is decoded to run from LINK9, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.5.15 LINK10_CFG Register (Offset = 58h) [Reset = 00000002h]

LINK10_CFG is shown in Figure 10-140 and described in Table 10-157.

Return to the Summary Table.

LINK10 Configuration

Figure 10-140. LINK10_CFG Register

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

Table 10-157. LINK10_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK10. When code is decoded to run from LINK10, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.5.16 LINK11_CFG Register (Offset = 5Ch) [Reset = 00000002h]

LINK11_CFG is shown in Figure 10-141 and described in Table 10-158.

Return to the Summary Table.

LINK11 Configuration

Figure 10-141. LINK11_CFG Register

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

Table 10-158. LINK11_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK11. When code is decoded to run from LINK11, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.5.17 LINK12_CFG Register (Offset = 60h) [Reset = 00000002h]

LINK12_CFG is shown in Figure 10-142 and described in Table 10-159.

Return to the Summary Table.

LINK12 Configuration

Figure 10-142. LINK12_CFG Register

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

Table 10-159. LINK12_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK12. When code is decoded to run from LINK12, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.5.18 LINK13_CFG Register (Offset = 64h) [Reset = 00000002h]

LINK13_CFG is shown in Figure 10-143 and described in Table 10-160.

Return to the Summary Table.

LINK13 Configuration

Figure 10-143. LINK13_CFG Register

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

Table 10-160. LINK13_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK13. When code is decoded to run from LINK13, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.5.19 LINK14_CFG Register (Offset = 68h) [Reset = 00000002h]

LINK14_CFG is shown in Figure 10-144 and described in Table 10-161.

Return to the Summary Table.

LINK14 Configuration

Figure 10-144. LINK14_CFG Register

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

Table 10-161. LINK14_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK14. When code is decoded to run from LINK14, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.5.20 LINK15_CFG Register (Offset = 6Ch) [Reset = 00000002h]

LINK15_CFG is shown in Figure 10-145 and described in Table 10-162.

Return to the Summary Table.

LINK15 Configuration

Figure 10-145. LINK15_CFG Register

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

Table 10-162. LINK15_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	STACK	R/W	2h	Defines the STACK association with LINK15. When code is decoded to run from LINK15, it will use the STACK defined in this register field. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : STACK1 0x2 : STACK2 (primary user STACK) ... Reset type: XRSn

10.16.5.21 STACK3_CFG Register (Offset = 7Ch) [Reset = 00000001h]

STACK3_CFG is shown in Figure 10-146 and described in Table 10-163.

Return to the Summary Table.

STACK3 Configuration

Figure 10-146. STACK3_CFG Register

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

Table 10-163. STACK3_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	ZONE	R/W	1h	Defines the ZONE association with STACK3. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : ZONE1 (primary user ZONE) 0x2 : ZONE2 (if available) 0x3 : ZONE3 (if available) Reset type: XRSn

10.16.5.22 STACK4_CFG Register (Offset = 80h) [Reset = 00000001h]

STACK4_CFG is shown in Figure 10-147 and described in Table 10-164.

Return to the Summary Table.

STACK4 Configuration

Figure 10-147. STACK4_CFG Register

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

Table 10-164. STACK4_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	ZONE	R/W	1h	Defines the ZONE association with STACK4. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : ZONE1 (primary user ZONE) 0x2 : ZONE2 (if available) 0x3 : ZONE3 (if available) Reset type: XRSn

10.16.5.23 STACK5_CFG Register (Offset = 84h) [Reset = 00000001h]

STACK5_CFG is shown in Figure 10-148 and described in Table 10-165.

Return to the Summary Table.

STACK5 Configuration

Figure 10-148. STACK5_CFG Register

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

Table 10-165. STACK5_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	ZONE	R/W	1h	Defines the ZONE association with STACK5. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : ZONE1 (primary user ZONE) 0x2 : ZONE2 (if available) 0x3 : ZONE3 (if available) Reset type: XRSn

10.16.5.24 STACK6_CFG Register (Offset = 88h) [Reset = 00000001h]

STACK6_CFG is shown in Figure 10-149 and described in Table 10-166.

Return to the Summary Table.

STACK6 Configuration

Figure 10-149. STACK6_CFG Register

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

Table 10-166. STACK6_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	ZONE	R/W	1h	Defines the ZONE association with STACK6. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : ZONE1 (primary user ZONE) 0x2 : ZONE2 (if available) 0x3 : ZONE3 (if available) Reset type: XRSn

10.16.5.25 STACK7_CFG Register (Offset = 8Ch) [Reset = 00000001h]

STACK7_CFG is shown in Figure 10-150 and described in Table 10-167.

Return to the Summary Table.

STACK7 Configuration

Figure 10-150. STACK7_CFG Register

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

Table 10-167. STACK7_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	ZONE	R/W	1h	Defines the ZONE association with STACK7. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : ZONE1 (primary user ZONE) 0x2 : ZONE2 (if available) 0x3 : ZONE3 (if available) Reset type: XRSn

10.16.5.26 RAMOPENSTAT Register (Offset = 90h) [Reset = 00000000h]

RAMOPENSTAT is shown in Figure 10-151 and described in Table 10-168.

Return to the Summary Table.

RAMOPEN Feature Status Register

Figure 10-151. RAMOPENSTAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						LINK1_RAMOPENS	RESERVED
R-0-0h						R-0h	R-0h

Table 10-168. RAMOPENSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	LINK1_RAMOPENS	R	0h	Reflects the status of the RAMOPEN feature for LINK1. 0 : RAMOPEN is not active. RAMs are assigned per the AP registers. 1 : RAMOPEN is active. Specified RAMs are assigned to LINK1. Reset type: XRSn
0	RESERVED	R	0h	Reserved

10.16.5.27 RAMOPENFRC Register (Offset = 94h) [Reset = 00000000h]

RAMOPENFRC is shown in Figure 10-152 and described in Table 10-169.

Return to the Summary Table.

RAMOPEN Feature Force Register

Figure 10-152. RAMOPENFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED			WIPE_ON_LINK1_CLR	RESERVED			SET_LINK1
R-0-0h			R/W1S-0h	R-0-0h			R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED			RESERVED	RESERVED			RESERVED
R-0-0h			R/W1S-0h	R-0-0h			R-0/W1S-0h

Table 10-169. RAMOPENFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-13	RESERVED	R-0	0h	Reserved
12	WIPE_ON_LINK1_CLR	R/W1S	0h	Defines if the HW goes through a RAMINIT when it closes RAMOPEN for LINK1. This bit is cleared after RAMOPEN is closed via RAMOPENCLR.CLEAR_LINK1. Reset type: XRSn
11-9	RESERVED	R-0	0h	Reserved
8	SET_LINK1	R-0/W1S	0h	Writing a '1' to this bit requests the 'RAMOPEN' mode for LINK1. When this bit is written with '1', hardware wipes out the RAM content using the RAMINIT feature and then sets the RAMOPENSTAT.RAMOPEN bit after completion of RAMINIT for all RAMOPEN-able RAMs. Writes of '0' are ignored. Reset type: XRSn
7-5	RESERVED	R-0	0h	Reserved
4	RESERVED	R/W1S	0h	Reserved
3-1	RESERVED	R-0	0h	Reserved
0	RESERVED	R-0/W1S	0h	Reserved

10.16.5.28 RAMOPENCLR Register (Offset = 98h) [Reset = 00000000h]

RAMOPENCLR is shown in Figure 10-153 and described in Table 10-170.

Return to the Summary Table.

RAMOPEN Feature Clear Register

Figure 10-153. RAMOPENCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						CLEAR_LINK1	RESERVED
R-0-0h						R-0/W1S-0h	R-0/W1S-0h

Table 10-170. RAMOPENCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-2	RESERVED	R-0	0h	Reserved
1	CLEAR_LINK1	R-0/W1S	0h	Writing a '1' to this bit clears the RAMOPENSTAT.LINK1_RAMOPENS bit, causing LINK1 to come out of 'RAMOPEN' mode. If RAMOPENFRC.WIPE_ON_LINK1_CLR is set, hardware wipes out the LINK1 RAMOPEN-able RAM content using the RAMINIT feature prior to clearing the RAMOPENSTAT.LINK1_RAMOPENS bit. RAMOPENSET.WIPE_ON_LINK1_CLR is also cleared when this bit is written with a '1'. Writes of '0' are ignored. Reset type: XRSn
0	RESERVED	R-0/W1S	0h	Reserved

10.16.5.29 DECODER_ADDR_IN Register (Offset = A0h) [Reset = 00000000h]

DECODER_ADDR_IN is shown in Figure 10-154 and described in Table 10-171.

Return to the Summary Table.

SW ZONE Decoder Address Input

Figure 10-154. DECODER_ADDR_IN Register

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

Table 10-171. DECODER_ADDR_IN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ADDR_H	R/W	0h	See ADDR_L. Reset type: XRSn
15-12	ADDR_L	R/W	0h	The 4KB address written to this register is evaluated against the user AP regions to determine the LINK, STACK, and ZONE of the address. The output is provided in DECODER_OUT. The Access Protection region number responding to the request is also provided. Reset type: XRSn
11-0	RESERVED	R	0h	Reserved

10.16.5.30 DECODER_OUT Register (Offset = A4h) [Reset = 80000000h]

DECODER_OUT is shown in Figure 10-155 and described in Table 10-172.

Return to the Summary Table.

SW ZONE Decoder Output

Figure 10-155. DECODER_OUT Register

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

Table 10-172. DECODER_OUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INVALID	R	1h	If either no AP region or multiple AP regions respond to the address query requested in DECODER_ADDR_IN, this bit will be set. If set, the contents in this register are not valid. Reset type: XRSn
30-23	RESERVED	R-0	0h	Reserved
22-16	APR	R	0h	See LINK. In SSUMODE1, this field will always read as a zero. Reset type: XRSn
15-10	RESERVED	R-0	0h	Reserved
9-8	ZONE	R	0h	See LINK. Reset type: XRSn
7	RESERVED	R-0	0h	Reserved
6-4	STACK	R	0h	See LINK. Reset type: XRSn
3-0	LINK	R	0h	The output from DECODER_ADDR_IN. Reset type: XRSn

10.16.5.31 EMU_DECODER_ADDR_IN Register (Offset = A8h) [Reset = 00000000h]

EMU_DECODER_ADDR_IN is shown in Figure 10-156 and described in Table 10-173.

Return to the Summary Table.

SW ZONE Decoder Address Input

Figure 10-156. EMU_DECODER_ADDR_IN Register

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

Table 10-173. EMU_DECODER_ADDR_IN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ADDR_H	R/W	0h	See ADDR_L. Reset type: XRSn
15-12	ADDR_L	R/W	0h	The 4KB address written to this register is evaluated against the user AP regions to determine the LINK, STACK, and ZONE of the address. The output is provided in DECODER_OUT. The Access Protection region number responding to the request is also provided. Debugger accesses should use this register and its corresponding output register and reserve the DECODER_ADDR_IN/OUT registers for runtime use. Reset type: XRSn
11-0	RESERVED	R	0h	Reserved

10.16.5.32 EMU_DECODER_OUT Register (Offset = ACh) [Reset = 80000000h]

EMU_DECODER_OUT is shown in Figure 10-157 and described in Table 10-174.

Return to the Summary Table.

SW ZONE Decoder Output

Figure 10-157. EMU_DECODER_OUT Register

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

Table 10-174. EMU_DECODER_OUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INVALID	R	1h	If either no AP region or multiple AP regions respond to the address query requested in EMU_DECODER_ADDR_IN, this bit will be set. If set, the contents in this register are not valid. Reset type: XRSn
30-23	RESERVED	R-0	0h	Reserved
22-16	APR	R	0h	See LINK. In SSUMODE1, this field will always read as a zero. Reset type: XRSn
15-10	RESERVED	R-0	0h	Reserved
9-8	ZONE	R	0h	See LINK. Reset type: XRSn
7	RESERVED	R-0	0h	Reserved
6-4	STACK	R	0h	See LINK. Reset type: XRSn
3-0	LINK	R	0h	The output from EMU_DECODER_ADDR_IN. Reset type: XRSn

10.16.6 SSU_CPU1_AP_REGS Registers

Table 10-175 lists the memory-mapped registers for the SSU_CPU1_AP_REGS registers. All register offset addresses not listed in Table 10-175 should be considered as reserved locations and the register contents should not be modified.

Table 10-175. SSU_CPU1_AP_REGS Registers

Offset	Acronym	Register Name	Protection
0h + formula	AP_CFG_j	Access Protection Configuration	
4h + formula	AP_START_EXT_j	Access Protection Start Address	
8h + formula	AP_END_EXT_j	Access Protection End Address	
Ch + formula	AP_LOCK_j	Access Protection Temporary Lock	
10h + formula	AP_COMMIT_j	Access Protection Permanent Commit	
14h + formula	AP_ACCESS_j	Access Protection R/W Access Permissions	
40h + formula	AP_CFG_j	Access Protection Configuration	
44h + formula	AP_START_j	Access Protection Start Address	
48h + formula	AP_END_j	Access Protection End Address	
4Ch + formula	AP_LOCK_j	Access Protection Temporary Lock	
50h + formula	AP_COMMIT_j	Access Protection Permanent Commit	
54h + formula	AP_ACCESS_j	Access Protection R/W Access Permissions	

Complex bit access types are encoded to fit into small table cells. Table 10-176 shows the codes that are used for access types in this section.

Table 10-176. SSU_CPU1_AP_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-*n*		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

10.16.6.1 AP_CFG_j Register (Offset = 0h + formula) [Reset = 00008242h]

AP_CFG_j is shown in Figure 10-158 and described in Table 10-177.

Return to the Summary Table.

Access Protection Configuration

Offset = 0h + (j * 20h); where j = 0h to 1h

Figure 10-158. AP_CFG_j Register

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

Table 10-177. AP_CFG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RESERVED	R/W	1h	Reserved
14	APILINKE	R/W	0h	Enables the Access Protection Inheritance LINK. 0 : Disabled 1 : Enabled Reset type: XRSn
13-12	RESERVED	R	0h	Reserved
11-8	APILINK	R/W	2h	If enabled, defines the common code LINK which inherits the permissions of its caller when making an access to this Access Protection's range. 0x0 : LINK0 0x1 : LINK1 (i.e., unsecure boot loaders) 0x2 : LINK2 (primary user LINK) 0x3 : LINK3 ... Reset type: XRSn
7	XE	R/W	0h	Enables execution from this Access Protection's range. 0 : Disabled 1 : Enabled Reset type: XRSn
6	APD	R/W	1h	Disables the AP{#} registers. If disabled, the AP region will not respond to any address range compares. 0 : Enabled 1 : Disabled Reset type: XRSn
5-4	RESERVED	R	0h	Reserved

Table 10-177. AP_CFG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	LINKID	R/W	2h	Defines the LINKID under which code from this region runs. Also used in determining which ZONE a target address belongs to during debug accesses as well as when performing Flash updates. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : LINK1 (i.e., unsecure boot loaders) 0x2 : LINK2 (primary user LINK) 0x3 : LINK3 ... Reset type: XRSn

10.16.6.2 AP_START_EXT_j Register (Offset = 4h + formula) [Reset = 00000000h]

AP_START_EXT_j is shown in Figure 10-159 and described in Table 10-178.

Return to the Summary Table.

Access Protection Start Address

Offset = 4h + (j * 20h); where j = 0h to 1h

Figure 10-159. AP_START_EXT_j Register

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

Table 10-178. AP_START_EXT_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ADDRH	R/W	0h	See ADDRL. Reset type: XRSn
15-12	ADDRL	R/W	0h	Extended address range start (4KB granularity, full address range). Reset type: XRSn
11-0	RESERVED	R	0h	Reserved

10.16.6.3 AP_END_EXT_j Register (Offset = 8h + formula) [Reset = 00000FFFh]

AP_END_EXT_j is shown in Figure 10-160 and described in Table 10-179.

Return to the Summary Table.

Access Protection End Address

Offset = 8h + (j * 20h); where j = 0h to 1h

Figure 10-160. AP_END_EXT_j Register

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

Table 10-179. AP_END_EXT_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ADDRH	R/W	0h	See ADDRL. Reset type: XRSn
15-12	ADDRL	R/W	0h	Extended address range end (4KB granularity, full address range). Reset type: XRSn
11-0	RESERVED	R	FFFh	Reserved

10.16.6.4 AP_LOCK_j Register (Offset = Ch + formula) [Reset = 00000000h]

AP_LOCK_j is shown in Figure 10-161 and described in Table 10-180.

Return to the Summary Table.

Access Protection Temporary Lock

Offset = Ch + (j * 20h); where j = 0h to 1h

Figure 10-161. AP_LOCK_j Register

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

Table 10-180. AP_LOCK_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks this Access Protection's registers (writes will have no effect on them). This bit can only be modified if AP{#}_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: XRSn

10.16.6.5 AP_COMMIT_j Register (Offset = 10h + formula) [Reset = 00000000h]

AP_COMMIT_j is shown in Figure 10-162 and described in Table 10-181.

Return to the Summary Table.

Access Protection Permanent Commit

Offset = 10h + (j * 20h); where j = 0h to 1h

Figure 10-162. AP_COMMIT_j Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 10-181. AP_COMMIT_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the AP{#}_LOCK register. This bit cannot be cleared, except by reset. 0 : AP{#}_LOCK is modifiable 1 : AP{#}_LOCK is committed permanently Reset type: XRSn

10.16.6.6 AP_ACCESS_j Register (Offset = 14h + formula) [Reset = 00000000h]

AP_ACCESS_j is shown in Figure 10-163 and described in Table 10-182.

Return to the Summary Table.

Access Protection R/W Access Permissions

Offset = 14h + (j * 20h); where j = 0h to 1h

Figure 10-163. AP_ACCESS_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LINK15		LINK14		LINK13		LINK12		LINK11		LINK10		LINK9		LINK8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK7		LINK6		LINK5		LINK4		LINK3		LINK2		LINK1		LINK0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 10-182. AP_ACCESS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	LINK15	R/W	0h	Enables code running with the LINKID of 15 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
29-28	LINK14	R/W	0h	Enables code running with the LINKID of 14 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
27-26	LINK13	R/W	0h	Enables code running with the LINKID of 13 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
25-24	LINK12	R/W	0h	Enables code running with the LINKID of 12 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
23-22	LINK11	R/W	0h	Enables code running with the LINKID of 11 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

Table 10-182. AP_ACCESS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21-20	LINK10	R/W	0h	Enables code running with the LINKID of 10 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
19-18	LINK9	R/W	0h	Enables code running with the LINKID of 9 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
17-16	LINK8	R/W	0h	Enables code running with the LINKID of 8 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
15-14	LINK7	R/W	0h	Enables code running with the LINKID of 7 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
13-12	LINK6	R/W	0h	Enables code running with the LINKID of 6 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
11-10	LINK5	R/W	0h	Enables code running with the LINKID of 5 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
9-8	LINK4	R/W	0h	Enables code running with the LINKID of 4 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
7-6	LINK3	R/W	0h	Enables code running with the LINKID of 3 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

Table 10-182. AP_ACCESS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	LINK2	R/W	0h	Enables code running with the LINKID of 2 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
3-2	LINK1	R/W	0h	Enables code running with the LINKID of 1 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
1-0	LINK0	R/W	0h	Enables code running with the LINKID of 0 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

10.16.6.7 AP_CFG_j Register (Offset = 40h + formula) [Reset = 00008242h]

AP_CFG_j is shown in Figure 10-164 and described in Table 10-183.

Return to the Summary Table.

Access Protection Configuration

Offset = 40h + (j * 20h); where j = 0h to 3Dh

Figure 10-164. AP_CFG_j Register

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

Table 10-183. AP_CFG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RESERVED	R/W	1h	Reserved
14	APILINKE	R/W	0h	Enables the Access Protection Inheritance LINK. 0 : Disabled 1 : Enabled Reset type: XRSn
13-12	RESERVED	R	0h	Reserved
11-8	APILINK	R/W	2h	If enabled, defines the common code LINK which inherits the permissions of its caller when making an access to this Access Protection's range. 0x0 : LINK0 0x1 : LINK1 (i.e., unsecure boot loaders) 0x2 : LINK2 (primary user LINK) 0x3 : LINK3 ... Reset type: XRSn
7	XE	R/W	0h	Enables execution from this Access Protection's range. 0 : Disabled 1 : Enabled Reset type: XRSn
6	APD	R/W	1h	Disables the AP{#} registers. If disabled, the AP region will not respond to any address range compares. 0 : Enabled 1 : Disabled Reset type: XRSn
5-4	RESERVED	R	0h	Reserved

Table 10-183. AP_CFG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	LINKID	R/W	2h	Defines the LINKID under which code from this region runs. Also used in determining which ZONE a target address belongs to during debug accesses as well as when performing Flash updates. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : LINK1 (i.e., unsecure boot loaders) 0x2 : LINK2 (primary user LINK) 0x3 : LINK3 ... Reset type: XRSn

10.16.6.8 AP_START_j Register (Offset = 44h + formula) [Reset = 00000000h]

AP_START_j is shown in Figure 10-165 and described in Table 10-184.

Return to the Summary Table.

Access Protection Start Address

Offset = 44h + (j * 20h); where j = 0h to 3Dh

Figure 10-165. AP_START_j Register

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

Table 10-184. AP_START_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30-28	MEMTYPE	R/W	0h	The part of the address range of the memory map that defines the memory type. Each type is a 256MB range, covering the first 2GB of total device memory range. 000 : ROM memory range 001 : Flash memory range 010 : RAM memory range 011 - 111: Peripherals Reset type: XRSn
27-24	RESERVED	R	0h	Reserved
23-16	ADDRH	R/W	0h	See ADDRL. Max 16MB range for each memory type. Reset type: XRSn
15-12	ADDRL	R/W	0h	Address range start address (granularity of 4KB) within the memory type specified. The values available depends on the memory type chosen and the memory footprint of the device. Reset type: XRSn
11-0	RESERVED	R	0h	Reserved

10.16.6.9 AP_END_j Register (Offset = 48h + formula) [Reset = 00000FFFh]

AP_END_j is shown in Figure 10-166 and described in Table 10-185.

Return to the Summary Table.

Access Protection End Address

Offset = 48h + (j * 20h); where j = 0h to 3Dh

Figure 10-166. AP_END_j Register

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

Table 10-185. AP_END_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30-28	MEMTYPE	R	0h	Reflects the state of AP{#}_START.MEMTYPE. Reset type: XRSn
27-24	RESERVED	R	0h	Reserved
23-16	ADDRH	R/W	0h	See ADDRL. Reset type: XRSn
15-12	ADDRL	R/W	0h	Address range end address (granularity of 4KB) within the memory type specified in AP{#}_START. Reset type: XRSn
11-0	RESERVED	R	FFFh	Reserved

10.16.6.10 AP_LOCK_j Register (Offset = 4Ch + formula) [Reset = 00000000h]

AP_LOCK_j is shown in Figure 10-167 and described in Table 10-186.

Return to the Summary Table.

Access Protection Temporary Lock

Offset = 4Ch + (j * 20h); where j = 0h to 3Dh

Figure 10-167. AP_LOCK_j Register

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

Table 10-186. AP_LOCK_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks this Access Protection's registers (writes will have no effect on them). This bit can only be modified if AP{#}_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: XRSn

10.16.6.11 AP_COMMIT_j Register (Offset = 50h + formula) [Reset = 00000000h]

AP_COMMIT_j is shown in Figure 10-168 and described in Table 10-187.

Return to the Summary Table.

Access Protection Permanent Commit

Offset = 50h + (j * 20h); where j = 0h to 3Dh

Figure 10-168. AP_COMMIT_j Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 10-187. AP_COMMIT_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the AP{#}_LOCK register. This bit cannot be cleared, except by reset. 0 : AP{#}_LOCK is modifiable 1 : AP{#}_LOCK is committed permanently Reset type: XRSn

10.16.6.12 AP_ACCESS_j Register (Offset = 54h + formula) [Reset = 00000000h]

AP_ACCESS_j is shown in Figure 10-169 and described in Table 10-188.

Return to the Summary Table.

Access Protection R/W Access Permissions

Offset = 54h + (j * 20h); where j = 0h to 3Dh

Figure 10-169. AP_ACCESS_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LINK15		LINK14		LINK13		LINK12		LINK11		LINK10		LINK9		LINK8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK7		LINK6		LINK5		LINK4		LINK3		LINK2		LINK1		LINK0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 10-188. AP_ACCESS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	LINK15	R/W	0h	Enables code running with the LINKID of 15 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
29-28	LINK14	R/W	0h	Enables code running with the LINKID of 14 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
27-26	LINK13	R/W	0h	Enables code running with the LINKID of 13 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
25-24	LINK12	R/W	0h	Enables code running with the LINKID of 12 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
23-22	LINK11	R/W	0h	Enables code running with the LINKID of 11 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

Table 10-188. AP_ACCESS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21-20	LINK10	R/W	0h	Enables code running with the LINKID of 10 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
19-18	LINK9	R/W	0h	Enables code running with the LINKID of 9 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
17-16	LINK8	R/W	0h	Enables code running with the LINKID of 8 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
15-14	LINK7	R/W	0h	Enables code running with the LINKID of 7 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
13-12	LINK6	R/W	0h	Enables code running with the LINKID of 6 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
11-10	LINK5	R/W	0h	Enables code running with the LINKID of 5 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
9-8	LINK4	R/W	0h	Enables code running with the LINKID of 4 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
7-6	LINK3	R/W	0h	Enables code running with the LINKID of 3 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

Table 10-188. AP_ACCESS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	LINK2	R/W	0h	Enables code running with the LINKID of 2 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
3-2	LINK1	R/W	0h	Enables code running with the LINKID of 1 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
1-0	LINK0	R/W	0h	Enables code running with the LINKID of 0 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

10.16.7 SSU_CPU2_AP_REGS Registers

Table 10-189 lists the memory-mapped registers for the SSU_CPU2_AP_REGS registers. All register offset addresses not listed in Table 10-189 should be considered as reserved locations and the register contents should not be modified.

Table 10-189. SSU_CPU2_AP_REGS Registers

Offset	Acronym	Register Name	Protection
0h + formula	AP_CFG_j	Access Protection Configuration	
4h + formula	AP_START_EXT_j	Access Protection Start Address	
8h + formula	AP_END_EXT_j	Access Protection End Address	
Ch + formula	AP_LOCK_j	Access Protection Temporary Lock	
10h + formula	AP_COMMIT_j	Access Protection Permanent Commit	
14h + formula	AP_ACCESS_j	Access Protection R/W Access Permissions	
40h + formula	AP_CFG_j	Access Protection Configuration	
44h + formula	AP_START_j	Access Protection Start Address	
48h + formula	AP_END_j	Access Protection End Address	
4Ch + formula	AP_LOCK_j	Access Protection Temporary Lock	
50h + formula	AP_COMMIT_j	Access Protection Permanent Commit	
54h + formula	AP_ACCESS_j	Access Protection R/W Access Permissions	

Complex bit access types are encoded to fit into small table cells. Table 10-190 shows the codes that are used for access types in this section.

Table 10-190. SSU_CPU2_AP_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-*n*		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

10.16.7.1 AP_CFG_j Register (Offset = 0h + formula) [Reset = 00008242h]

AP_CFG_j is shown in Figure 10-170 and described in Table 10-191.

Return to the Summary Table.

Access Protection Configuration

Offset = 0h + (j * 20h); where j = 0h to 1h

Figure 10-170. AP_CFG_j Register

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

Table 10-191. AP_CFG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RESERVED	R/W	1h	Reserved
14	APILINKE	R/W	0h	Enables the Access Protection Inheritance LINK. 0 : Disabled 1 : Enabled Reset type: XRSn
13-12	RESERVED	R	0h	Reserved
11-8	APILINK	R/W	2h	If enabled, defines the common code LINK which inherits the permissions of its caller when making an access to this Access Protection's range. 0x0 : LINK0 0x1 : LINK1 (i.e., unsecure boot loaders) 0x2 : LINK2 (primary user LINK) 0x3 : LINK3 ... Reset type: XRSn
7	XE	R/W	0h	Enables execution from this Access Protection's range. 0 : Disabled 1 : Enabled Reset type: XRSn
6	APD	R/W	1h	Disables the AP{#} registers. If disabled, the AP region will not respond to any address range compares. 0 : Enabled 1 : Disabled Reset type: XRSn
5-4	RESERVED	R	0h	Reserved

Table 10-191. AP_CFG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	LINKID	R/W	2h	Defines the LINKID under which code from this region runs. Also used in determining which ZONE a target address belongs to during debug accesses as well as when performing Flash updates. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : LINK1 (i.e., unsecure boot loaders) 0x2 : LINK2 (primary user LINK) 0x3 : LINK3 ... Reset type: XRSn

10.16.7.2 AP_START_EXT_j Register (Offset = 4h + formula) [Reset = 00000000h]

AP_START_EXT_j is shown in Figure 10-171 and described in Table 10-192.

Return to the Summary Table.

Access Protection Start Address

Offset = 4h + (j * 20h); where j = 0h to 1h

Figure 10-171. AP_START_EXT_j Register

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

Table 10-192. AP_START_EXT_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ADDRH	R/W	0h	See ADDRL. Reset type: XRSn
15-12	ADDRL	R/W	0h	Extended address range start (4KB granularity, full address range). Reset type: XRSn
11-0	RESERVED	R	0h	Reserved

10.16.7.3 AP_END_EXT_j Register (Offset = 8h + formula) [Reset = 00000FFFh]

AP_END_EXT_j is shown in Figure 10-172 and described in Table 10-193.

Return to the Summary Table.

Access Protection End Address

Offset = 8h + (j * 20h); where j = 0h to 1h

Figure 10-172. AP_END_EXT_j Register

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

Table 10-193. AP_END_EXT_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ADDRH	R/W	0h	See ADDRL. Reset type: XRSn
15-12	ADDRL	R/W	0h	Extended address range end (4KB granularity, full address range). Reset type: XRSn
11-0	RESERVED	R	FFFh	Reserved

10.16.7.4 AP_LOCK_j Register (Offset = Ch + formula) [Reset = 00000000h]

AP_LOCK_j is shown in Figure 10-173 and described in Table 10-194.

Return to the Summary Table.

Access Protection Temporary Lock

Offset = Ch + (j * 20h); where j = 0h to 1h

Figure 10-173. AP_LOCK_j Register

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

Table 10-194. AP_LOCK_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks this Access Protection's registers (writes will have no effect on them). This bit can only be modified if AP{#}_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: XRSn

10.16.7.5 AP_COMMIT_j Register (Offset = 10h + formula) [Reset = 00000000h]

AP_COMMIT_j is shown in Figure 10-174 and described in Table 10-195.

Return to the Summary Table.

Access Protection Permanent Commit

Offset = 10h + (j * 20h); where j = 0h to 1h

Figure 10-174. AP_COMMIT_j Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 10-195. AP_COMMIT_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the AP{#}_LOCK register. This bit cannot be cleared, except by reset. 0 : AP{#}_LOCK is modifiable 1 : AP{#}_LOCK is committed permanently Reset type: XRSn

10.16.7.6 AP_ACCESS_j Register (Offset = 14h + formula) [Reset = 00000000h]

AP_ACCESS_j is shown in Figure 10-175 and described in Table 10-196.

Return to the Summary Table.

Access Protection R/W Access Permissions

Offset = 14h + (j * 20h); where j = 0h to 1h

Figure 10-175. AP_ACCESS_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LINK15		LINK14		LINK13		LINK12		LINK11		LINK10		LINK9		LINK8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK7		LINK6		LINK5		LINK4		LINK3		LINK2		LINK1		LINK0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 10-196. AP_ACCESS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	LINK15	R/W	0h	Enables code running with the LINKID of 15 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
29-28	LINK14	R/W	0h	Enables code running with the LINKID of 14 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
27-26	LINK13	R/W	0h	Enables code running with the LINKID of 13 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
25-24	LINK12	R/W	0h	Enables code running with the LINKID of 12 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
23-22	LINK11	R/W	0h	Enables code running with the LINKID of 11 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

Table 10-196. AP_ACCESS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21-20	LINK10	R/W	0h	Enables code running with the LINKID of 10 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
19-18	LINK9	R/W	0h	Enables code running with the LINKID of 9 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
17-16	LINK8	R/W	0h	Enables code running with the LINKID of 8 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
15-14	LINK7	R/W	0h	Enables code running with the LINKID of 7 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
13-12	LINK6	R/W	0h	Enables code running with the LINKID of 6 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
11-10	LINK5	R/W	0h	Enables code running with the LINKID of 5 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
9-8	LINK4	R/W	0h	Enables code running with the LINKID of 4 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
7-6	LINK3	R/W	0h	Enables code running with the LINKID of 3 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

Table 10-196. AP_ACCESS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	LINK2	R/W	0h	Enables code running with the LINKID of 2 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
3-2	LINK1	R/W	0h	Enables code running with the LINKID of 1 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
1-0	LINK0	R/W	0h	Enables code running with the LINKID of 0 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

10.16.7.7 AP_CFG_j Register (Offset = 40h + formula) [Reset = 00008242h]

AP_CFG_j is shown in Figure 10-176 and described in Table 10-197.

Return to the Summary Table.

Access Protection Configuration

Offset = 40h + (j * 20h); where j = 0h to 3Dh

Figure 10-176. AP_CFG_j Register

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

Table 10-197. AP_CFG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RESERVED	R/W	1h	Reserved
14	APILINKE	R/W	0h	Enables the Access Protection Inheritance LINK. 0 : Disabled 1 : Enabled Reset type: XRSn
13-12	RESERVED	R	0h	Reserved
11-8	APILINK	R/W	2h	If enabled, defines the common code LINK which inherits the permissions of its caller when making an access to this Access Protection's range. 0x0 : LINK0 0x1 : LINK1 (i.e., unsecure boot loaders) 0x2 : LINK2 (primary user LINK) 0x3 : LINK3 ... Reset type: XRSn
7	XE	R/W	0h	Enables execution from this Access Protection's range. 0 : Disabled 1 : Enabled Reset type: XRSn
6	APD	R/W	1h	Disables the AP{#} registers. If disabled, the AP region will not respond to any address range compares. 0 : Enabled 1 : Disabled Reset type: XRSn
5-4	RESERVED	R	0h	Reserved

Table 10-197. AP_CFG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	LINKID	R/W	2h	Defines the LINKID under which code from this region runs. Also used in determining which ZONE a target address belongs to during debug accesses as well as when performing Flash updates. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : LINK1 (i.e., unsecure boot loaders) 0x2 : LINK2 (primary user LINK) 0x3 : LINK3 ... Reset type: XRSn

10.16.7.8 AP_START_j Register (Offset = 44h + formula) [Reset = 00000000h]

AP_START_j is shown in Figure 10-177 and described in Table 10-198.

Return to the Summary Table.

Access Protection Start Address

Offset = 44h + (j * 20h); where j = 0h to 3Dh

Figure 10-177. AP_START_j Register

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

Table 10-198. AP_START_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30-28	MEMTYPE	R/W	0h	The part of the address range of the memory map that defines the memory type. Each type is a 256MB range, covering the first 2GB of total device memory range. 000 : ROM memory range 001 : Flash memory range 010 : RAM memory range 011 - 111: Peripherals Reset type: XRSn
27-24	RESERVED	R	0h	Reserved
23-16	ADDRH	R/W	0h	See ADDRL. Max 16MB range for each memory type. Reset type: XRSn
15-12	ADDRL	R/W	0h	Address range start address (granularity of 4KB) within the memory type specified. The values available depends on the memory type chosen and the memory footprint of the device. Reset type: XRSn
11-0	RESERVED	R	0h	Reserved

10.16.7.9 AP_END_j Register (Offset = 48h + formula) [Reset = 00000FFFh]

AP_END_j is shown in Figure 10-178 and described in Table 10-199.

Return to the Summary Table.

Access Protection End Address

Offset = 48h + (j * 20h); where j = 0h to 3Dh

Figure 10-178. AP_END_j Register

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

Table 10-199. AP_END_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30-28	MEMTYPE	R	0h	Reflects the state of AP{#}_START.MEMTYPE. Reset type: XRSn
27-24	RESERVED	R	0h	Reserved
23-16	ADDRH	R/W	0h	See ADDRL. Reset type: XRSn
15-12	ADDRL	R/W	0h	Address range end address (granularity of 4KB) within the memory type specified in AP{#}_START. Reset type: XRSn
11-0	RESERVED	R	FFFh	Reserved

10.16.7.10 AP_LOCK_j Register (Offset = 4Ch + formula) [Reset = 00000000h]

AP_LOCK_j is shown in Figure 10-179 and described in Table 10-200.

Return to the Summary Table.

Access Protection Temporary Lock

Offset = 4Ch + (j * 20h); where j = 0h to 3Dh

Figure 10-179. AP_LOCK_j Register

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

Table 10-200. AP_LOCK_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks this Access Protection's registers (writes will have no effect on them). This bit can only be modified if AP{#}_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: XRSn

10.16.7.11 AP_COMMIT_j Register (Offset = 50h + formula) [Reset = 00000000h]

AP_COMMIT_j is shown in Figure 10-180 and described in Table 10-201.

Return to the Summary Table.

Access Protection Permanent Commit

Offset = 50h + (j * 20h); where j = 0h to 3Dh

Figure 10-180. AP_COMMIT_j Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 10-201. AP_COMMIT_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the AP{#}_LOCK register. This bit cannot be cleared, except by reset. 0 : AP{#}_LOCK is modifiable 1 : AP{#}_LOCK is committed permanently Reset type: XRSn

10.16.7.12 AP_ACCESS_j Register (Offset = 54h + formula) [Reset = 00000000h]

AP_ACCESS_j is shown in Figure 10-181 and described in Table 10-202.

Return to the Summary Table.

Access Protection R/W Access Permissions

Offset = 54h + (j * 20h); where j = 0h to 3Dh

Figure 10-181. AP_ACCESS_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LINK15		LINK14		LINK13		LINK12		LINK11		LINK10		LINK9		LINK8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK7		LINK6		LINK5		LINK4		LINK3		LINK2		LINK1		LINK0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 10-202. AP_ACCESS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	LINK15	R/W	0h	Enables code running with the LINKID of 15 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
29-28	LINK14	R/W	0h	Enables code running with the LINKID of 14 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
27-26	LINK13	R/W	0h	Enables code running with the LINKID of 13 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
25-24	LINK12	R/W	0h	Enables code running with the LINKID of 12 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
23-22	LINK11	R/W	0h	Enables code running with the LINKID of 11 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

Table 10-202. AP_ACCESS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21-20	LINK10	R/W	0h	Enables code running with the LINKID of 10 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
19-18	LINK9	R/W	0h	Enables code running with the LINKID of 9 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
17-16	LINK8	R/W	0h	Enables code running with the LINKID of 8 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
15-14	LINK7	R/W	0h	Enables code running with the LINKID of 7 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
13-12	LINK6	R/W	0h	Enables code running with the LINKID of 6 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
11-10	LINK5	R/W	0h	Enables code running with the LINKID of 5 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
9-8	LINK4	R/W	0h	Enables code running with the LINKID of 4 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
7-6	LINK3	R/W	0h	Enables code running with the LINKID of 3 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

Table 10-202. AP_ACCESS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	LINK2	R/W	0h	Enables code running with the LINKID of 2 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
3-2	LINK1	R/W	0h	Enables code running with the LINKID of 1 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
1-0	LINK0	R/W	0h	Enables code running with the LINKID of 0 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

10.16.8 SSU_CPU3_AP_REGS Registers

Table 10-203 lists the memory-mapped registers for the SSU_CPU3_AP_REGS registers. All register offset addresses not listed in Table 10-203 should be considered as reserved locations and the register contents should not be modified.

Table 10-203. SSU_CPU3_AP_REGS Registers

Offset	Acronym	Register Name	Protection
0h + formula	AP_CFG_j	Access Protection Configuration	
4h + formula	AP_START_EXT_j	Access Protection Start Address	
8h + formula	AP_END_EXT_j	Access Protection End Address	
Ch + formula	AP_LOCK_j	Access Protection Temporary Lock	
10h + formula	AP_COMMIT_j	Access Protection Permanent Commit	
14h + formula	AP_ACCESS_j	Access Protection R/W Access Permissions	
40h + formula	AP_CFG_j	Access Protection Configuration	
44h + formula	AP_START_j	Access Protection Start Address	
48h + formula	AP_END_j	Access Protection End Address	
4Ch + formula	AP_LOCK_j	Access Protection Temporary Lock	
50h + formula	AP_COMMIT_j	Access Protection Permanent Commit	
54h + formula	AP_ACCESS_j	Access Protection R/W Access Permissions	

Complex bit access types are encoded to fit into small table cells. Table 10-204 shows the codes that are used for access types in this section.

Table 10-204. SSU_CPU3_AP_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-*n*		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

10.16.8.1 AP_CFG_j Register (Offset = 0h + formula) [Reset = 00008242h]

AP_CFG_j is shown in Figure 10-182 and described in Table 10-205.

Return to the Summary Table.

Access Protection Configuration

Offset = 0h + (j * 20h); where j = 0h to 1h

Figure 10-182. AP_CFG_j Register

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

Table 10-205. AP_CFG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RESERVED	R/W	1h	Reserved
14	APILINKE	R/W	0h	Enables the Access Protection Inheritance LINK. 0 : Disabled 1 : Enabled Reset type: XRSn
13-12	RESERVED	R	0h	Reserved
11-8	APILINK	R/W	2h	If enabled, defines the common code LINK which inherits the permissions of its caller when making an access to this Access Protection's range. 0x0 : LINK0 0x1 : LINK1 (i.e., unsecure boot loaders) 0x2 : LINK2 (primary user LINK) 0x3 : LINK3 ... Reset type: XRSn
7	XE	R/W	0h	Enables execution from this Access Protection's range. 0 : Disabled 1 : Enabled Reset type: XRSn
6	APD	R/W	1h	Disables the AP{#} registers. If disabled, the AP region will not respond to any address range compares. 0 : Enabled 1 : Disabled Reset type: XRSn
5-4	RESERVED	R	0h	Reserved

Table 10-205. AP_CFG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	LINKID	R/W	2h	Defines the LINKID under which code from this region runs. Also used in determining which ZONE a target address belongs to during debug accesses as well as when performing Flash updates. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : LINK1 (i.e., unsecure boot loaders) 0x2 : LINK2 (primary user LINK) 0x3 : LINK3 ... Reset type: XRSn

10.16.8.2 AP_START_EXT_j Register (Offset = 4h + formula) [Reset = 00000000h]

AP_START_EXT_j is shown in Figure 10-183 and described in Table 10-206.

Return to the Summary Table.

Access Protection Start Address

Offset = 4h + (j * 20h); where j = 0h to 1h

Figure 10-183. AP_START_EXT_j Register

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

Table 10-206. AP_START_EXT_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ADDRH	R/W	0h	See ADDRL. Reset type: XRSn
15-12	ADDRL	R/W	0h	Extended address range start (4KB granularity, full address range). Reset type: XRSn
11-0	RESERVED	R	0h	Reserved

10.16.8.3 AP_END_EXT_j Register (Offset = 8h + formula) [Reset = 00000FFFh]

AP_END_EXT_j is shown in Figure 10-184 and described in Table 10-207.

Return to the Summary Table.

Access Protection End Address

Offset = 8h + (j * 20h); where j = 0h to 1h

Figure 10-184. AP_END_EXT_j Register

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

Table 10-207. AP_END_EXT_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ADDRH	R/W	0h	See ADDRL. Reset type: XRSn
15-12	ADDRL	R/W	0h	Extended address range end (4KB granularity, full address range). Reset type: XRSn
11-0	RESERVED	R	FFFh	Reserved

10.16.8.4 AP_LOCK_j Register (Offset = Ch + formula) [Reset = 00000000h]

AP_LOCK_j is shown in Figure 10-185 and described in Table 10-208.

Return to the Summary Table.

Access Protection Temporary Lock

Offset = Ch + (j * 20h); where j = 0h to 1h

Figure 10-185. AP_LOCK_j Register

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

Table 10-208. AP_LOCK_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks this Access Protection's registers (writes will have no effect on them). This bit can only be modified if AP{#}_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: XRSn

10.16.8.5 AP_COMMIT_j Register (Offset = 10h + formula) [Reset = 00000000h]

AP_COMMIT_j is shown in Figure 10-186 and described in Table 10-209.

Return to the Summary Table.

Access Protection Permanent Commit

Offset = 10h + (j * 20h); where j = 0h to 1h

Figure 10-186. AP_COMMIT_j Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 10-209. AP_COMMIT_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the AP{#}_LOCK register. This bit cannot be cleared, except by reset. 0 : AP{#}_LOCK is modifiable 1 : AP{#}_LOCK is committed permanently Reset type: XRSn

10.16.8.6 AP_ACCESS_j Register (Offset = 14h + formula) [Reset = 00000000h]

AP_ACCESS_j is shown in Figure 10-187 and described in Table 10-210.

Return to the Summary Table.

Access Protection R/W Access Permissions

Offset = 14h + (j * 20h); where j = 0h to 1h

Figure 10-187. AP_ACCESS_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LINK15		LINK14		LINK13		LINK12		LINK11		LINK10		LINK9		LINK8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK7		LINK6		LINK5		LINK4		LINK3		LINK2		LINK1		LINK0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 10-210. AP_ACCESS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	LINK15	R/W	0h	Enables code running with the LINKID of 15 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
29-28	LINK14	R/W	0h	Enables code running with the LINKID of 14 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
27-26	LINK13	R/W	0h	Enables code running with the LINKID of 13 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
25-24	LINK12	R/W	0h	Enables code running with the LINKID of 12 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
23-22	LINK11	R/W	0h	Enables code running with the LINKID of 11 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

Table 10-210. AP_ACCESS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21-20	LINK10	R/W	0h	Enables code running with the LINKID of 10 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
19-18	LINK9	R/W	0h	Enables code running with the LINKID of 9 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
17-16	LINK8	R/W	0h	Enables code running with the LINKID of 8 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
15-14	LINK7	R/W	0h	Enables code running with the LINKID of 7 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
13-12	LINK6	R/W	0h	Enables code running with the LINKID of 6 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
11-10	LINK5	R/W	0h	Enables code running with the LINKID of 5 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
9-8	LINK4	R/W	0h	Enables code running with the LINKID of 4 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
7-6	LINK3	R/W	0h	Enables code running with the LINKID of 3 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

Table 10-210. AP_ACCESS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	LINK2	R/W	0h	Enables code running with the LINKID of 2 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
3-2	LINK1	R/W	0h	Enables code running with the LINKID of 1 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
1-0	LINK0	R/W	0h	Enables code running with the LINKID of 0 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

10.16.8.7 AP_CFG_j Register (Offset = 40h + formula) [Reset = 00008242h]

AP_CFG_j is shown in Figure 10-188 and described in Table 10-211.

Return to the Summary Table.

Access Protection Configuration

Offset = 40h + (j * 20h); where j = 0h to 3Dh

Figure 10-188. AP_CFG_j Register

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

Table 10-211. AP_CFG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RESERVED	R/W	1h	Reserved
14	APILINKE	R/W	0h	Enables the Access Protection Inheritance LINK. 0 : Disabled 1 : Enabled Reset type: XRSn
13-12	RESERVED	R	0h	Reserved
11-8	APILINK	R/W	2h	If enabled, defines the common code LINK which inherits the permissions of its caller when making an access to this Access Protection's range. 0x0 : LINK0 0x1 : LINK1 (i.e., unsecure boot loaders) 0x2 : LINK2 (primary user LINK) 0x3 : LINK3 ... Reset type: XRSn
7	XE	R/W	0h	Enables execution from this Access Protection's range. 0 : Disabled 1 : Enabled Reset type: XRSn
6	APD	R/W	1h	Disables the AP{#} registers. If disabled, the AP region will not respond to any address range compares. 0 : Enabled 1 : Disabled Reset type: XRSn
5-4	RESERVED	R	0h	Reserved

Table 10-211. AP_CFG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	LINKID	R/W	2h	Defines the LINKID under which code from this region runs. Also used in determining which ZONE a target address belongs to during debug accesses as well as when performing Flash updates. 0x0 : Reserved (a write attempt of 0x0 sets value to the HW reset value) 0x1 : LINK1 (i.e., unsecure boot loaders) 0x2 : LINK2 (primary user LINK) 0x3 : LINK3 ... Reset type: XRSn

10.16.8.8 AP_START_j Register (Offset = 44h + formula) [Reset = 00000000h]

AP_START_j is shown in Figure 10-189 and described in Table 10-212.

Return to the Summary Table.

Access Protection Start Address

Offset = 44h + (j * 20h); where j = 0h to 3Dh

Figure 10-189. AP_START_j Register

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

Table 10-212. AP_START_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30-28	MEMTYPE	R/W	0h	The part of the address range of the memory map that defines the memory type. Each type is a 256MB range, covering the first 2GB of total device memory range. 000 : ROM memory range 001 : Flash memory range 010 : RAM memory range 011 - 111: Peripherals Reset type: XRSn
27-24	RESERVED	R	0h	Reserved
23-16	ADDRH	R/W	0h	See ADDRL. Max 16MB range for each memory type. Reset type: XRSn
15-12	ADDRL	R/W	0h	Address range start address (granularity of 4KB) within the memory type specified. The values available depends on the memory type chosen and the memory footprint of the device. Reset type: XRSn
11-0	RESERVED	R	0h	Reserved

10.16.8.9 AP_END_j Register (Offset = 48h + formula) [Reset = 00000FFFh]

AP_END_j is shown in Figure 10-190 and described in Table 10-213.

Return to the Summary Table.

Access Protection End Address

Offset = 48h + (j * 20h); where j = 0h to 3Dh

Figure 10-190. AP_END_j Register

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

Table 10-213. AP_END_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30-28	MEMTYPE	R	0h	Reflects the state of AP{#}_START.MEMTYPE. Reset type: XRSn
27-24	RESERVED	R	0h	Reserved
23-16	ADDRH	R/W	0h	See ADDRL. Reset type: XRSn
15-12	ADDRL	R/W	0h	Address range end address (granularity of 4KB) within the memory type specified in AP{#}_START. Reset type: XRSn
11-0	RESERVED	R	FFFh	Reserved

10.16.8.10 AP_LOCK_j Register (Offset = 4Ch + formula) [Reset = 00000000h]

AP_LOCK_j is shown in Figure 10-191 and described in Table 10-214.

Return to the Summary Table.

Access Protection Temporary Lock

Offset = 4Ch + (j * 20h); where j = 0h to 3Dh

Figure 10-191. AP_LOCK_j Register

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

Table 10-214. AP_LOCK_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks this Access Protection's registers (writes will have no effect on them). This bit can only be modified if AP{#}_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: XRSn

10.16.8.11 AP_COMMIT_j Register (Offset = 50h + formula) [Reset = 00000000h]

AP_COMMIT_j is shown in Figure 10-192 and described in Table 10-215.

Return to the Summary Table.

Access Protection Permanent Commit

Offset = 50h + (j * 20h); where j = 0h to 3Dh

Figure 10-192. AP_COMMIT_j Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 10-215. AP_COMMIT_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the AP{#}_LOCK register. This bit cannot be cleared, except by reset. 0 : AP{#}_LOCK is modifiable 1 : AP{#}_LOCK is committed permanently Reset type: XRSn

10.16.8.12 AP_ACCESS_j Register (Offset = 54h + formula) [Reset = 00000000h]

AP_ACCESS_j is shown in Figure 10-193 and described in Table 10-216.

Return to the Summary Table.

Access Protection R/W Access Permissions

Offset = 54h + (j * 20h); where j = 0h to 3Dh

Figure 10-193. AP_ACCESS_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LINK15		LINK14		LINK13		LINK12		LINK11		LINK10		LINK9		LINK8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK7		LINK6		LINK5		LINK4		LINK3		LINK2		LINK1		LINK0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 10-216. AP_ACCESS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	LINK15	R/W	0h	Enables code running with the LINKID of 15 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
29-28	LINK14	R/W	0h	Enables code running with the LINKID of 14 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
27-26	LINK13	R/W	0h	Enables code running with the LINKID of 13 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
25-24	LINK12	R/W	0h	Enables code running with the LINKID of 12 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
23-22	LINK11	R/W	0h	Enables code running with the LINKID of 11 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

Table 10-216. AP_ACCESS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21-20	LINK10	R/W	0h	Enables code running with the LINKID of 10 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
19-18	LINK9	R/W	0h	Enables code running with the LINKID of 9 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
17-16	LINK8	R/W	0h	Enables code running with the LINKID of 8 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
15-14	LINK7	R/W	0h	Enables code running with the LINKID of 7 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
13-12	LINK6	R/W	0h	Enables code running with the LINKID of 6 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
11-10	LINK5	R/W	0h	Enables code running with the LINKID of 5 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
9-8	LINK4	R/W	0h	Enables code running with the LINKID of 4 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
7-6	LINK3	R/W	0h	Enables code running with the LINKID of 3 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

Table 10-216. AP_ACCESS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	LINK2	R/W	0h	Enables code running with the LINKID of 2 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
3-2	LINK1	R/W	0h	Enables code running with the LINKID of 1 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn
1-0	LINK0	R/W	0h	Enables code running with the LINKID of 0 to access this Access Protection's range. 00 : No access 01 : Read Access 10 : Reserved 11 : Read/Write Access Reset type: XRSn

Chapter 11

Configurable Logic Block (CLB)

This chapter describes the features and operation of the configurable logic block (CLB) that is a collection of configurable blocks that can be inter-connected using software to implement custom digital logic functions.

11.1 Introduction

The configurable logic block (CLB) is a collection of configurable blocks that can be inter-connected using software to implement custom digital logic functions. The CLB is able to enhance existing peripherals through a set of crossbar interconnections, which provide a high level of connectivity to existing control peripherals such as enhanced pulse width modulators (ePWM), enhanced capture modules (eCAP), and enhanced quadrature encoder pulse modules (eQEP). The crossbars also allow the CLB to be connected to external GPIO pins. In this way, the CLB can be configured to interact with device peripherals to perform small logical functions such as simple PWM generators, or to implement custom serial data exchange protocols.

The CLB peripheral is configured through the CLB tool. For more information on the CLB tool, available examples, application reports, and user's guide, refer to the following location in your C2000WARE package (C2000Ware_2_00_00_03 and higher): C2000WARE_INSTALL_LOCATION\utilities\clb_tool\clb_syscfg\doc

11.1.1 CLB Related Collateral

Foundational Materials

- C2000™ Configurable Logic Block (CLB) Series (Video)
- C28x Academy - CLB
- C29x Academy - Configurable Logic Block (CLB)
- Customizing on-chip peripherals defies conventional logic
- *Enable Differentiation and win with CLB in various applications Application Report*
- Enable Differentiation with Configurable Logic in Various Automotive Applications (Video)

Getting Started Materials

- *C2000™ Position Manager PTO API Reference Guide Application Report*
- *CLB Tool User Guide*
  - Basic examples are 7 - 15 (start with these). More involved examples are 1-6.
- *Designing With The C2000 Configurable Logic Block Application Report*
- *How to Migrate Custom Logic From an FPGA/CPLD to C2000 Microcontrollers Application Report*
  - Chpaters 1-3 are very useful for getting started and learning the CLB. Later chapters are very useful Expert materials for migrating from FPGA/CPLD to C2000's CLB.

Expert Materials

- *Achieve Delayed Protection for Three-Level Inverter With CLB Application Report*
- *Diagnosing Delta-Sigma Modulator Bitstream Using C2000™ Configurable Logic Block Application Report*
- *How to Implement Custom Serial Interfaces Using Configurable Logic Block (CLB) Application Report*
- Tamagawa T-Format Absolute-Encoder Master Interface Reference Design for C2000™ MCUs

11.2 Description

The CLB subsystem contains a number of identical tiles. There are four such tiles in the CLB subsystem; other devices can contain more or fewer tiles. Tiles are numbered 1 to N, where N is the total tile count on the device. Each tile contains combinational and sequential logic blocks, as well as other dedicated hardware to be described later in this document. Figure 11-1 shows the structure of the CLB subsystem in the device.

The tile contains the core logic, providing the logic reconfiguration capability. Each CLB tile is associated with a separate CPU interface, which contains the registers needed to control and configure the logic in the tile. The CPU interface also contains data transfer buffers that can be used as part of the configurable logic to exchange data with the rest of the device. Figure 11-2 shows the connections between the tile, the CPU interface, and the device.

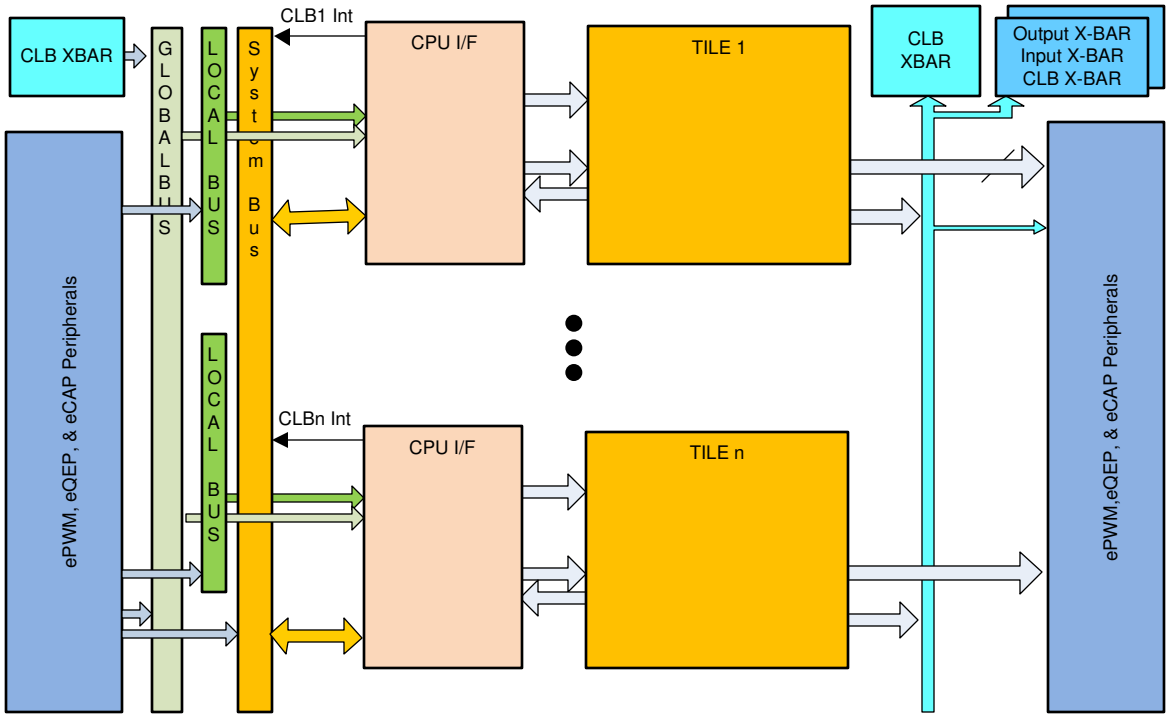


Figure 11-1. Block Diagram of the CLB Subsystem in the Device

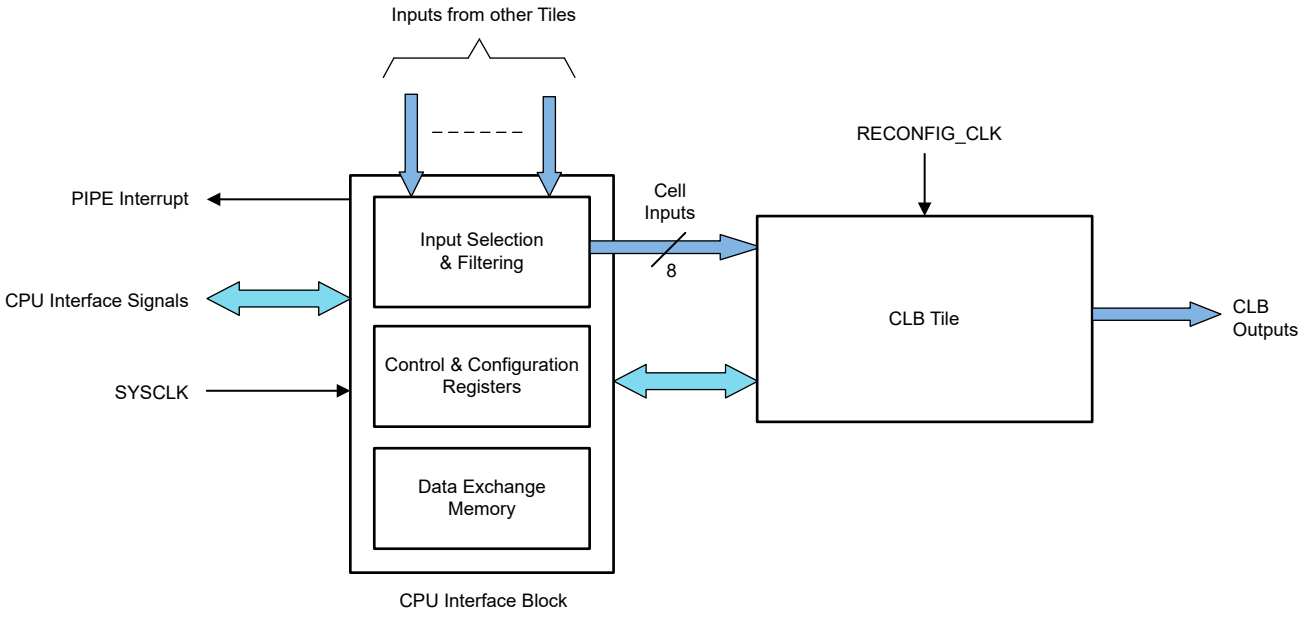


Figure 11-2. Block Diagram of a CLB Tile and CPU Interface

11.2.1 CLB Clock

In this device, the CLB clock is called CLBx clock that can be enabled or disabled by SYSCTL_PERIPH_CLK_CLBx through the SysCtl_enablePeripheral function. The maximum frequency is 150MHz and the clock can be enabled and configured by modifying the CLBx clock. Refer to the device data sheet for specifications on frequencies for the CLB. The *Clocking System* figure in the *System Control and Interrupts* chapter shows how the CLB is clocked.

Note

When clock frequencies are above 100MHz (for example, 150MHz), PIPELINE mode must be enabled.

The CLB TILE clock and CLB register clock can be in ASYNC/SYNC mode with the SYSCLK. An example CLB clock configuration is shown in Table 11-1. Check the device data sheet for details on clocking specifications.

Table 11-1. Example CLB Clocking Configuration

Clock	SYNC Mode (CLKMODECLBx = 0)	ASYNC Mode (CLKMODECLBx = 1)	
		TILECLKDIV = 1	TILECLKDIV = 0
CLB Register Clock	SYSCLK	SYSCLK	SYSCLK
CLB TILE Clock	SYSCLK	SYSCLK / 2	SYSCLK

Starting with CLB Type 2, a clock prescalar module is available. The prescalar module can generate a prescaled version of the CLB clock signal that can be used as an input to the CLB TILE's counter.

Note

The prescalar logic does not change the actual clocking speed of the CLB. The prescalar generates a strobe (that can toggle at the defined prescaled rate) that is made available as another input signal to the CLB logic and the strobe is only used when required.

The prescalar module is shown in Figure 11-3.

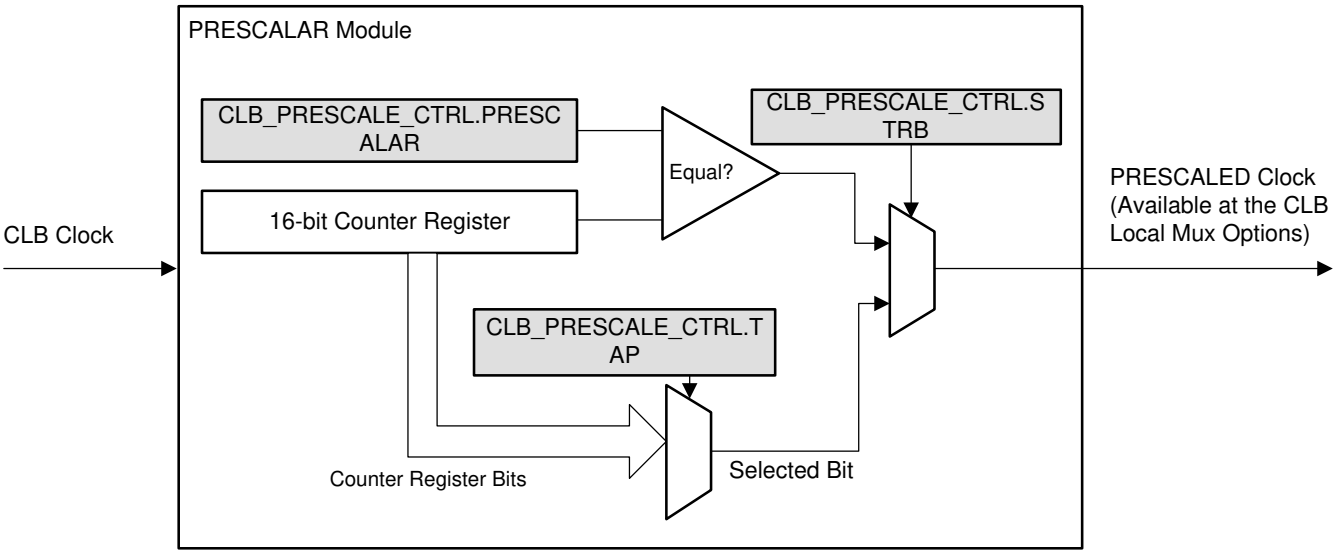


Figure 11-3. CLB Clock Prescalar

11.3 CLB Input/Output Connection

11.3.1 Overview

There are four instances of the CLB module in the device. Each CLB instance has a common set of input signals referred to as global input signals. Additionally, each CLB instance has a specific set of input signals that are unique to each instance, and are referred to as local input signals. Each of the eight inputs of a CLB can be chosen from any of the global input signals or the local input signals.

Note

Signals routed into the CLB using the XBAR must be synchronized within the CLB.

11.3.2 CLB Input Selection

Each CLB module has eight inputs that are applied to the reconfigurable logic cell. Each of these inputs can be selectively driven by a predefined set of signals. A two-level mux structure allows each input of each CLB instance to select a signal.

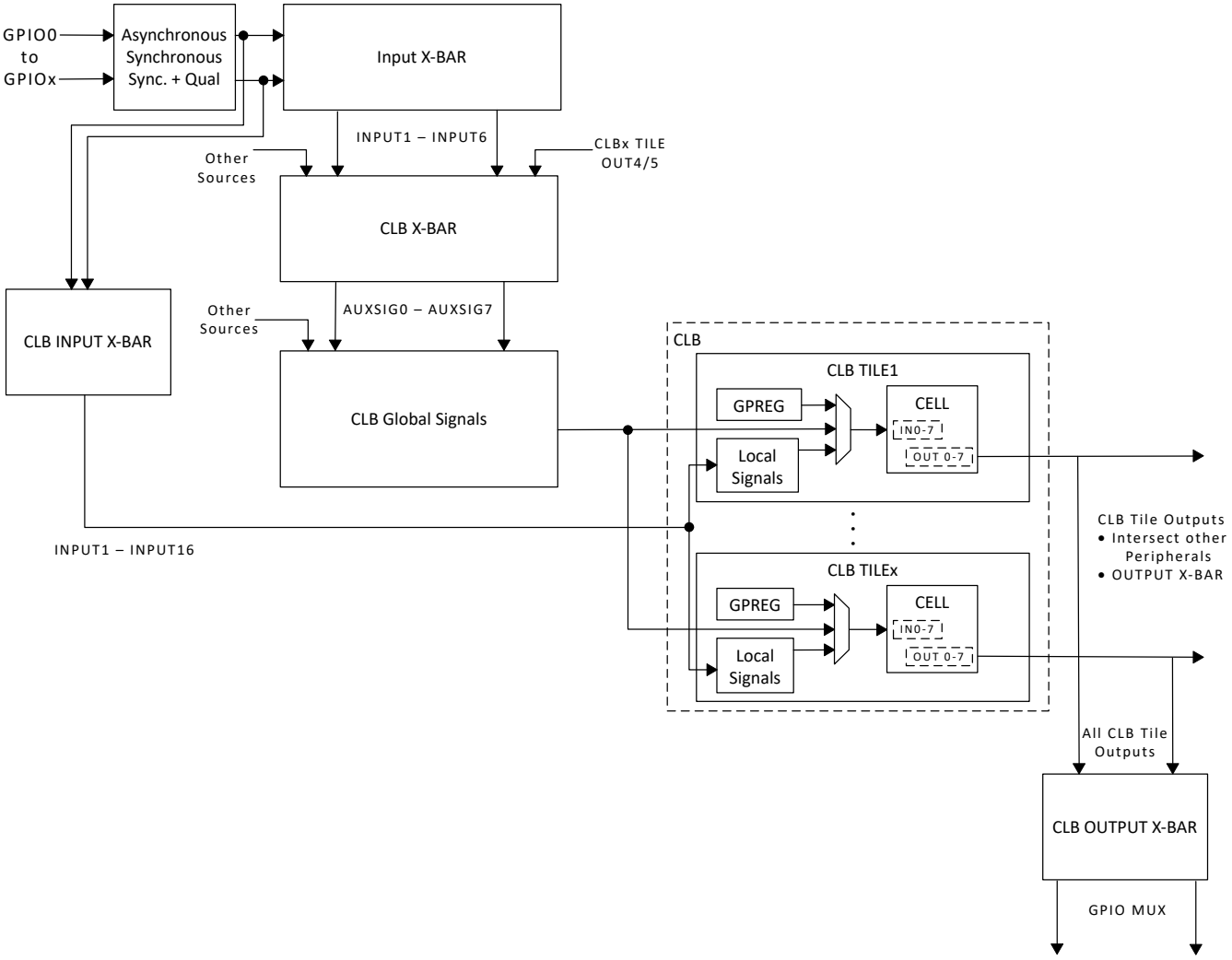


Figure 11-4. GPIO to CLB Tile Connections

A set of signals is common to all the CLB instances. These are referred to as global inputs in Figure 11-5. A separate set of signals is unique to each instance of the CLB. These are referred to as local inputs in Figure 11-5.

Registers CLB_LCL_MUX_SEL_1 and CLB_LCL_MUX_SEL_2 control the local mux selection for each of the eight inputs. The mux control registers CLB_GLBL_MUX_SEL_1 and CLB_GLBL_MUX_SEL_2 control the global mux selection for each of the eight inputs.

The local mux select value of 0 causes the selected global mux input signal to be connected to the corresponding CLB Input. For example, setting CLB_LCL_MUX_SEL_IN_0 = 0 and CLB_GLBL_MUX_SEL_IN_0 = 8 causes the global mux input number 8 to be connected to CLB Input 0. The input filter feature can be used to enable edge detection on the CLB inputs. The input filter feature can also synchronize the input with the CLB clock.

The global mux settings are shown in Table 11-2 and Table 11-3 . The local input mux settings are shown in Table 11-4 and Table 11-5 .

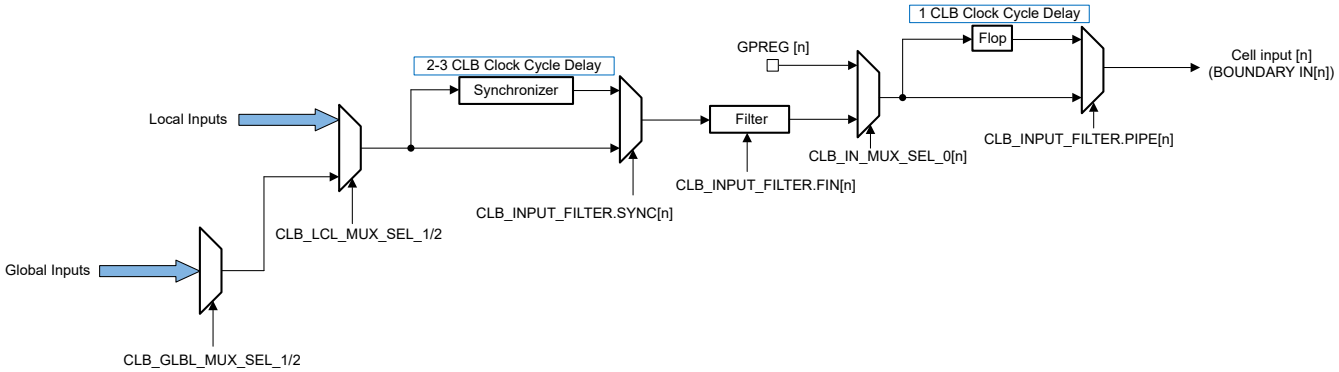


Figure 11-5. CLB Input Mux and Filter

Figure 11-6 shows an example of how to use synchronization for an asynchronous signal, in this case the ePWM signal. Figure 11-7 shows an instance of using input pipelining for a synchronous signal, which here is the ePWM TBCLK signal. Note that these two input configurations are not used simultaneously, and each have a cycle delay that adds to the input path.

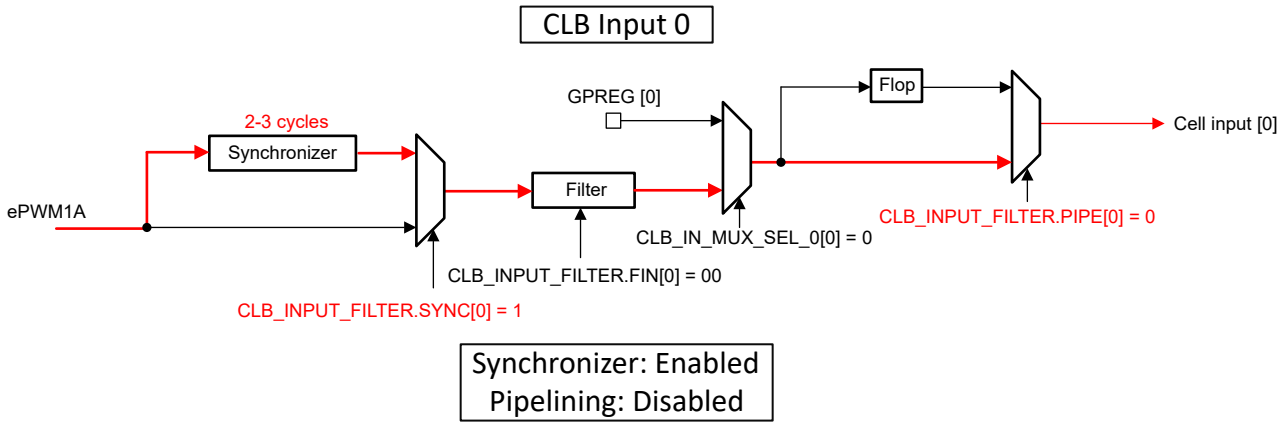


Figure 11-6. CLB Input Synchronization Example

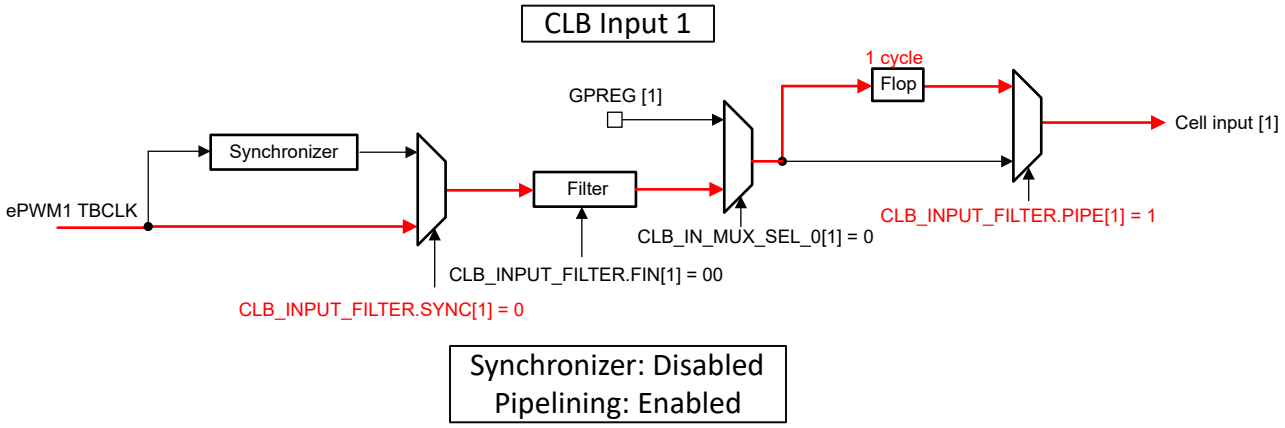


Figure 11-7. CLB Input Pipelining Example

Note

If a signal in the following table indicates that synchronization is required, then the CLB input synchronizer must be enabled using the appropriate SYNC bit in the CLB_INPUT_FILTER register. This synchronization adds a 2-3 CLB clock cycle delay to the input. This delay is either 2 or 3 cycles and is not predictable. There is a potential for a metastability hazard, if the indicated signals are not first synchronized before going into the CLB tile. This metastability can cause errors dependent on voltage, temperature, and wafer fab process. Note that this requirement is in addition to and separate from GPIO input synchronization.

If a signal in the following table indicates that synchronization is not required, as the signal is already synchronous, then pipelining is required and must be enabled using the PIPE bit in the CLB_INPUT_FILTER register. This pipelining adds a 1 CLB clock cycle delay to the input. This is not to be mistaken with the PIPELINE_EN bit in the CLB_LOAD_EN register, which controls pipelining of the CLB operations in the HLC and counter blocks. This PIPELINE_EN bit is also used when the device is run above 100MHz. Having synchronization and pipelining both enabled or both disabled is not recommended. Enabling both synchronization and pipelining introduces a delay of more than 2-3 CLB clock cycles on the signal path. Disabling both allows the completely asynchronous signal to be routed as an input.

Table 11-2. Global Signals and Mux Selection

Select Value	CLB1 Input	CLB2 Input	CLB3 Input	CLB4 Input	Synchronization Requirement
0	EPWM1A	EPWM1A	EPWM1A	EPWM1A	Enable
1	EPWM1A_OE	EPWM1A_OE	EPWM1A_OE	EPWM1A_OE	Enable
2	EPWM1B	EPWM1B	EPWM1B	EPWM1B	Enable
3	EPWM1B_OE	EPWM1B_OE	EPWM1B_OE	EPWM1B_OE	Enable
4	EPWM1_CTR_ZERO	EPWM1_CTR_ZERO	EPWM1_CTR_ZERO	EPWM1_CTR_ZERO	Disable
5	EPWM1_CTR_PRD	EPWM1_CTR_PRD	EPWM1_CTR_PRD	EPWM1_CTR_PRD	Disable
6	EPWM1_CTR_DIR	EPWM1_CTR_DIR	EPWM1_CTR_DIR	EPWM1_CTR_DIR	Disable
7	EPWM1_TBCLK	EPWM1_TBCLK	EPWM1_TBCLK	EPWM1_TBCLK	Disable
8	EPWM1_CTR_CMPA	EPWM1_CTR_CMPA	EPWM1_CTR_CMPA	EPWM1_CTR_CMPA	Disable
9	EPWM1_CTR_CMPB	EPWM1_CTR_CMPB	EPWM1_CTR_CMPB	EPWM1_CTR_CMPB	Disable
10	EPWM1_CTR_CMPC	EPWM1_CTR_CMPC	EPWM1_CTR_CMPC	EPWM1_CTR_CMPC	Disable

Table 11-2. Global Signals and Mux Selection (continued)

Select Value	CLB1 Input	CLB2 Input	CLB3 Input	CLB4 Input	Synchronization Requirement
11	EPWM1_CTR_CMPD	EPWM1_CTR_CMPD	EPWM1_CTR_CMPD	EPWM1_CTR_CMPD	Disable
12	EPWM1A_AQ	EPWM1A_AQ	EPWM1A_AQ	EPWM1A_AQ	Disable
13	EPWM1B_AQ	EPWM1B_AQ	EPWM1B_AQ	EPWM1B_AQ	Disable
14	EPWM1A_DB	EPWM1A_DB	EPWM1A_DB	EPWM1A_DB	Enable
15	EPWM1B_DB	EPWM1B_DB	EPWM1B_DB	EPWM1B_DB	Enable
16	EPWM2A	EPWM2A	EPWM2A	EPWM2A	Enable
17	EPWM2A_OE	EPWM2A_OE	EPWM2A_OE	EPWM2A_OE	Enable
18	EPWM2B	EPWM2B	EPWM2B	EPWM2B	Enable
19	EPWM2B_OE	EPWM2B_OE	EPWM2B_OE	EPWM2B_OE	Enable
20	EPWM2_CTR_ZERO	EPWM2_CTR_ZERO	EPWM2_CTR_ZERO	EPWM2_CTR_ZERO	Disable
21	EPWM2_CTR_PRD	EPWM2_CTR_PRD	EPWM2_CTR_PRD	EPWM2_CTR_PRD	Disable
22	EPWM2_CTR_DIR	EPWM2_CTR_DIR	EPWM2_CTR_DIR	EPWM2_CTR_DIR	Disable
23	EPWM2_TBCLK	EPWM2_TBCLK	EPWM2_TBCLK	EPWM2_TBCLK	Disable
24	EPWM2_CTR_CMPA	EPWM2_CTR_CMPA	EPWM2_CTR_CMPA	EPWM2_CTR_CMPA	Disable
25	EPWM2_CTR_CMPB	EPWM2_CTR_CMPB	EPWM2_CTR_CMPB	EPWM2_CTR_CMPB	Disable
26	EPWM2_CTR_CMPC	EPWM2_CTR_CMPC	EPWM2_CTR_CMPC	EPWM2_CTR_CMPC	Disable
27	EPWM2_CTR_CMPD	EPWM2_CTR_CMPD	EPWM2_CTR_CMPD	EPWM2_CTR_CMPD	Disable
28	EPWM2A_AQ	EPWM2A_AQ	EPWM2A_AQ	EPWM2A_AQ	Disable
29	EPWM2B_AQ	EPWM2B_AQ	EPWM2B_AQ	EPWM2B_AQ	Disable
30	EPWM2A_DB	EPWM2A_DB	EPWM2A_DB	EPWM2A_DB	Enable
31	EPWM2B_DB	EPWM2B_DB	EPWM2B_DB	EPWM2B_DB	Enable
32	EPWM3A	EPWM3A	EPWM3A	EPWM3A	Enable
33	EPWM3A_OE	EPWM3A_OE	EPWM3A_OE	EPWM3A_OE	Enable
34	EPWM3B	EPWM3B	EPWM3B	EPWM3B	Enable
35	EPWM3B_OE	EPWM3B_OE	EPWM3B_OE	EPWM3B_OE	Enable
36	EPWM3_CTR_ZERO	EPWM3_CTR_ZERO	EPWM3_CTR_ZERO	EPWM3_CTR_ZERO	Disable
37	EPWM3_CTR_PRD	EPWM3_CTR_PRD	EPWM3_CTR_PRD	EPWM3_CTR_PRD	Disable
38	EPWM3_CTR_DIR	EPWM3_CTR_DIR	EPWM3_CTR_DIR	EPWM3_CTR_DIR	Disable
39	EPWM3_TBCLK	EPWM3_TBCLK	EPWM3_TBCLK	EPWM3_TBCLK	Disable
40	EPWM3_CTR_CMPA	EPWM3_CTR_CMPA	EPWM3_CTR_CMPA	EPWM3_CTR_CMPA	Disable
41	EPWM3_CTR_CMPB	EPWM3_CTR_CMPB	EPWM3_CTR_CMPB	EPWM3_CTR_CMPB	Disable
42	EPWM3_CTR_CMPC	EPWM3_CTR_CMPC	EPWM3_CTR_CMPC	EPWM3_CTR_CMPC	Disable
43	EPWM3_CTR_CMPD	EPWM3_CTR_CMPD	EPWM3_CTR_CMPD	EPWM3_CTR_CMPD	Disable
44	EPWM3A_AQ	EPWM3A_AQ	EPWM3A_AQ	EPWM3A_AQ	Disable
45	EPWM3B_AQ	EPWM3B_AQ	EPWM3B_AQ	EPWM3B_AQ	Disable
46	EPWM3A_DB	EPWM3A_DB	EPWM3A_DB	EPWM3A_DB	Enable
47	EPWM3B_DB	EPWM3B_DB	EPWM3B_DB	EPWM3B_DB	Enable

Table 11-2. Global Signals and Mux Selection (continued)

Select Value	CLB1 Input	CLB2 Input	CLB3 Input	CLB4 Input	Synchronization Requirement
48	EPWM4A	EPWM4A	EPWM4A	EPWM4A	Enable
49	EPWM4A_OE	EPWM4A_OE	EPWM4A_OE	EPWM4A_OE	Enable
50	EPWM4B	EPWM4B	EPWM4B	EPWM4B	Enable
51	EPWM4B_OE	EPWM4B_OE	EPWM4B_OE	EPWM4B_OE	Enable
52	EPWM4_CTR_ZERO	EPWM4_CTR_ZERO	EPWM4_CTR_ZERO	EPWM4_CTR_ZERO	Disable
53	EPWM4_CTR_PRD	EPWM4_CTR_PRD	EPWM4_CTR_PRD	EPWM4_CTR_PRD	Disable
54	EPWM4_CTR_DIR	EPWM4_CTR_DIR	EPWM4_CTR_DIR	EPWM4_CTR_DIR	Disable
55	EPWM4_TBCLK	EPWM4_TBCLK	EPWM4_TBCLK	EPWM4_TBCLK	Disable
56	EPWM4_CTR_CMPA	EPWM4_CTR_CMPA	EPWM4_CTR_CMPA	EPWM4_CTR_CMPA	Disable
57	EPWM4_CTR_CMPB	EPWM4_CTR_CMPB	EPWM4_CTR_CMPB	EPWM4_CTR_CMPB	Disable
58	EPWM4_CTR_CMPC	EPWM4_CTR_CMPC	EPWM4_CTR_CMPC	EPWM4_CTR_CMPC	Disable
59	EPWM4_CTR_CMPD	EPWM4_CTR_CMPD	EPWM4_CTR_CMPD	EPWM4_CTR_CMPD	Disable
60	EPWM4A_AQ	EPWM4A_AQ	EPWM4A_AQ	EPWM4A_AQ	Disable
61	EPWM4B_AQ	EPWM4B_AQ	EPWM4B_AQ	EPWM4B_AQ	Disable
62	EPWM4A_DB	EPWM4A_DB	EPWM4A_DB	EPWM4A_DB	Enable
63	EPWM4B_DB	EPWM4B_DB	EPWM4B_DB	EPWM4B_DB	Enable
64	AUXSIG0	AUXSIG0	AUXSIG0	AUXSIG0	Enable
65	AUXSIG1	AUXSIG1	AUXSIG1	AUXSIG1	Enable
66	AUXSIG2	AUXSIG2	AUXSIG2	AUXSIG2	Enable
67	AUXSIG3	AUXSIG3	AUXSIG3	AUXSIG3	Enable
68	AUXSIG4	AUXSIG4	AUXSIG4	AUXSIG4	Enable
69	AUXSIG5	AUXSIG5	AUXSIG5	AUXSIG5	Enable
70	AUXSIG6	AUXSIG6	AUXSIG6	AUXSIG6	Enable
71	AUXSIG7	AUXSIG7	AUXSIG7	AUXSIG7	Enable
72	CLB1_OUT16	CLB1_OUT16	CLB1_OUT16	CLB1_OUT16	Disable
73	CLB1_OUT17	CLB1_OUT17	CLB1_OUT17	CLB1_OUT17	Disable
74	CLB1_OUT18	CLB1_OUT18	CLB1_OUT18	CLB1_OUT18	Disable
75	CLB1_OUT19	CLB1_OUT19	CLB1_OUT19	CLB1_OUT19	Disable
76	CLB1_OUT20	CLB1_OUT20	CLB1_OUT20	CLB1_OUT20	Disable
77	CLB1_OUT21	CLB1_OUT21	CLB1_OUT21	CLB1_OUT21	Disable
78	CLB1_OUT22	CLB1_OUT22	CLB1_OUT22	CLB1_OUT22	Disable
79	CLB1_OUT23	CLB1_OUT23	CLB1_OUT23	CLB1_OUT23	Disable
80	CLB2_OUT16	CLB2_OUT16	CLB2_OUT16	CLB2_OUT16	Disable
81	CLB2_OUT17	CLB2_OUT17	CLB2_OUT17	CLB2_OUT17	Disable
82	CLB2_OUT18	CLB2_OUT18	CLB2_OUT18	CLB2_OUT18	Disable
83	CLB2_OUT19	CLB2_OUT19	CLB2_OUT19	CLB2_OUT19	Disable
84	CLB2_OUT20	CLB2_OUT20	CLB2_OUT20	CLB2_OUT20	Disable

Table 11-2. Global Signals and Mux Selection (continued)

Select Value	CLB1 Input	CLB2 Input	CLB3 Input	CLB4 Input	Synchronization Requirement
85	CLB2_OUT21	CLB2_OUT21	CLB2_OUT21	CLB2_OUT21	Disable
86	CLB2_OUT22	CLB2_OUT22	CLB2_OUT22	CLB2_OUT22	Disable
87	CLB2_OUT23	CLB2_OUT23	CLB2_OUT23	CLB2_OUT23	Disable
88	CLB3_OUT16	CLB3_OUT16	CLB3_OUT16	CLB3_OUT16	Disable
89	CLB3_OUT17	CLB3_OUT17	CLB3_OUT17	CLB3_OUT17	Disable
90	CLB3_OUT18	CLB3_OUT18	CLB3_OUT18	CLB3_OUT18	Disable
91	CLB3_OUT19	CLB3_OUT19	CLB3_OUT19	CLB3_OUT19	Disable
92	CLB3_OUT20	CLB3_OUT20	CLB3_OUT20	CLB3_OUT20	Disable
93	CLB3_OUT21	CLB3_OUT21	CLB3_OUT21	CLB3_OUT21	Disable
94	CLB3_OUT22	CLB3_OUT22	CLB3_OUT22	CLB3_OUT22	Disable
95	CLB3_OUT23	CLB3_OUT23	CLB3_OUT23	CLB3_OUT23	Disable
96	CLB4_OUT16	CLB4_OUT16	CLB4_OUT16	CLB4_OUT16	Disable
97	CLB4_OUT17	CLB4_OUT17	CLB4_OUT17	CLB4_OUT17	Disable
98	CLB4_OUT18	CLB4_OUT18	CLB4_OUT18	CLB4_OUT18	Disable
99	CLB4_OUT19	CLB4_OUT19	CLB4_OUT19	CLB4_OUT19	Disable
100	CLB4_OUT20	CLB4_OUT20	CLB4_OUT20	CLB4_OUT20	Disable
101	CLB4_OUT21	CLB4_OUT21	CLB4_OUT21	CLB4_OUT21	Disable
102	CLB4_OUT22	CLB4_OUT22	CLB4_OUT22	CLB4_OUT22	Disable
103	CLB4_OUT23	CLB4_OUT23	CLB4_OUT23	CLB4_OUT23	Disable
104	CPU1_ERAD_EBC_EVT0	CPU2_ERAD_EBC_EVT0	CPU1_ERAD_SEC_EVT0	CPU3_ERAD_EBC_EVT0	Disable
105	CPU1_ERAD_EBC_EVT1	CPU2_ERAD_EBC_EVT1	CPU1_ERAD_SEC_EVT1	CPU3_ERAD_EBC_EVT1	Disable
106	CPU1_ERAD_EBC_EVT2	CPU2_ERAD_EBC_EVT2	CPU1_ERAD_SEC_EVT2	CPU3_ERAD_EBC_EVT2	Disable
107	CPU1_ERAD_EBC_EVT3	CPU2_ERAD_EBC_EVT3	CPU1_ERAD_SEC_EVT3	CPU3_ERAD_EBC_EVT3	Disable
108	CPU1_ERAD_EBC_EVT4	CPU2_ERAD_EBC_EVT4	CPU2_ERAD_SEC_EVT0	CPU3_ERAD_EBC_EVT4	Disable
109	CPU1_ERAD_EBC_EVT5	CPU2_ERAD_EBC_EVT5	CPU2_ERAD_SEC_EVT1	CPU3_ERAD_EBC_EVT5	Disable
110	CPU1_ERAD_EBC_EVT6	CPU2_ERAD_EBC_EVT6	CPU2_ERAD_SEC_EVT2	CPU3_ERAD_EBC_EVT6	Disable
111	CPU1_ERAD_EBC_EVT7	CPU2_ERAD_EBC_EVT7	CPU2_ERAD_SEC_EVT3	CPU3_ERAD_EBC_EVT7	Disable
112	FSIRXA_DATA_PKT_RCVD	FSIRXA_DATA_PKT_RCVD	FSIRXA_DATA_PKT_RCVD	FSIRXA_DATA_PKT_RCVD	Disable
113	FSIRXA_ERROR_PKT_RCVD	FSIRXA_ERROR_PKT_RCVD	FSIRXA_ERROR_PKT_RCVD	FSIRXA_ERROR_PKT_RCVD	Disable

Table 11-2. Global Signals and Mux Selection (continued)

Select Value	CLB1 Input	CLB2 Input	CLB3 Input	CLB4 Input	Synchronization Requirement
114	FSIRXA_PING_PKT_RCVD	FSIRXA_PING_PKT_RCVD	FSIRXA_PING_PKT_RCVD	FSIRXA_PING_PKT_RCVD	Disable
115	FSIRXA_FRAME_DONE	FSIRXA_FRAME_DONE	FSIRXA_FRAME_DONE	FSIRXA_FRAME_DONE	Disable
116	FSIRXA_PING_TAG_MATCH	FSIRXA_PING_TAG_MATCH	FSIRXA_PING_TAG_MATCH	FSIRXA_PING_TAG_MATCH	Disable
117	FSIRXA_DATA_TAG_MATCH	FSIRXA_DATA_TAG_MATCH	FSIRXA_DATA_TAG_MATCH	FSIRXA_DATA_TAG_MATCH	Disable
118	FSIRXA_ERROR_TAG_MATCH	FSIRXA_ERROR_TAG_MATCH	FSIRXA_ERROR_TAG_MATCH	FSIRXA_ERROR_TAG_MATCH	Disable
119	FSIRXA_TRIG2	FSIRXA_TRIG2	FSIRXA_TRIG2	FSIRXA_TRIG2	Disable
120	SPIA_CLK_OUT	SPIA_CLK_OUT	SPIA_CLK_OUT	SPIA_CLK_OUT	Enable
121	SPIA_POCI_IN	SPIA_POCI_IN	SPIA_POCI_IN	SPIA_POCI_IN	Enable
122	SPIA_PTE_OUT	SPIA_PTE_OUT	SPIA_PTE_OUT	SPIA_PTE_OUT	Enable
123	SPIB_CLK_OUT	SPIB_CLK_OUT	SPIB_CLK_OUT	SPIB_CLK_OUT	Enable
124	SPIB_POCI_IN	SPIB_POCI_IN	SPIB_POCI_IN	SPIB_POCI_IN	Enable
125	SPIB_PTE_OUT	SPIB_PTE_OUT	SPIB_PTE_OUT	SPIB_PTE_OUT	Enable
126	CPU3_HALT	CPU3_HALT	CPU3_HALT	CPU3_HALT	Disable
127	FSIRXA_TRIG3	FSIRXA_TRIG3	FSIRXA_TRIG3	FSIRXA_TRIG3	Disable

Table 11-3. Global Signals and Mux Selection

Select Value	CLB5 Input	CLB6 Input	Synchronization Requirement
0	EPWM5A	EPWM5A	Enable
1	EPWM5A_OE	EPWM5A_OE	Enable
2	EPWM5B	EPWM5B	Enable
3	EPWM5B_OE	EPWM5B_OE	Enable
4	EPWM5_CTR_ZERO	EPWM5_CTR_ZERO	Disable
5	EPWM5_CTR_PRD	EPWM5_CTR_PRD	Disable
6	EPWM5_CTR_DIR	EPWM5_CTR_DIR	Disable
7	EPWM5_TBCLK	EPWM5_TBCLK	Disable
8	EPWM5_CTR_CMPA	EPWM5_CTR_CMPA	Disable
9	EPWM5_CTR_CMPB	EPWM5_CTR_CMPB	Disable
10	EPWM5_CTR_CMPC	EPWM5_CTR_CMPC	Disable
11	EPWM5_CTR_CMPD	EPWM5_CTR_CMPD	Disable
12	EPWM5A_AQ	EPWM5A_AQ	Disable
13	EPWM5B_AQ	EPWM5B_AQ	Disable
14	EPWM5A_DB	EPWM5A_DB	Enable
15	EPWM5B_DB	EPWM5B_DB	Enable
16	EPWM6A	EPWM6A	Enable

Table 11-3. Global Signals and Mux Selection (continued)

Select Value	CLB5 Input	CLB6 Input	Synchronization Requirement
17	EPWM6A_OE	EPWM6A_OE	Enable
18	EPWM6B	EPWM6B	Enable
19	EPWM6B_OE	EPWM6B_OE	Enable
20	EPWM6_CTR_ZERO	EPWM6_CTR_ZERO	Disable
21	EPWM6_CTR_PRD	EPWM6_CTR_PRD	Disable
22	EPWM6_CTR_DIR	EPWM6_CTR_DIR	Disable
23	EPWM6_TBCLK	EPWM6_TBCLK	Disable
24	EPWM6_CTR_CMPA	EPWM6_CTR_CMPA	Disable
25	EPWM6_CTR_CMPB	EPWM6_CTR_CMPB	Disable
26	EPWM6_CTR_CMPC	EPWM6_CTR_CMPC	Disable
27	EPWM6_CTR_CMPD	EPWM6_CTR_CMPD	Disable
28	EPWM6A_AQ	EPWM6A_AQ	Disable
29	EPWM6B_AQ	EPWM6B_AQ	Disable
30	EPWM6A_DB	EPWM6A_DB	Enable
31	EPWM6B_DB	EPWM6B_DB	Enable
32	EPWM7A	EPWM7A	Enable
33	EPWM7A_OE	EPWM7A_OE	Enable
34	EPWM7B	EPWM7B	Enable
35	EPWM7B_OE	EPWM7B_OE	Enable
36	EPWM7_CTR_ZERO	EPWM7_CTR_ZERO	Disable
37	EPWM7_CTR_PRD	EPWM7_CTR_PRD	Disable
38	EPWM7_CTR_DIR	EPWM7_CTR_DIR	Disable
39	EPWM7_TBCLK	EPWM7_TBCLK	Disable
40	EPWM7_CTR_CMPA	EPWM7_CTR_CMPA	Disable
41	EPWM7_CTR_CMPB	EPWM7_CTR_CMPB	Disable
42	EPWM7_CTR_CMPC	EPWM7_CTR_CMPC	Disable
43	EPWM7_CTR_CMPD	EPWM7_CTR_CMPD	Disable
44	EPWM7A_AQ	EPWM7A_AQ	Disable
45	EPWM7B_AQ	EPWM7B_AQ	Disable
46	EPWM7A_DB	EPWM7A_DB	Enable
47	EPWM7B_DB	EPWM7B_DB	Enable
48	EPWM8A	EPWM8A	Enable
49	EPWM8A_OE	EPWM8A_OE	Enable
50	EPWM8B	EPWM8B	Enable
51	EPWM8B_OE	EPWM8B_OE	Enable
52	EPWM8_CTR_ZERO	EPWM8_CTR_ZERO	Disable
53	EPWM8_CTR_PRD	EPWM8_CTR_PRD	Disable
54	EPWM8_CTR_DIR	EPWM8_CTR_DIR	Disable

Table 11-3. Global Signals and Mux Selection (continued)

Select Value	CLB5 Input	CLB6 Input	Synchronization Requirement
55	EPWM8_TBCLK	EPWM8_TBCLK	Disable
56	EPWM8_CTR_CMPA	EPWM8_CTR_CMPA	Disable
57	EPWM8_CTR_CMPB	EPWM8_CTR_CMPB	Disable
58	EPWM8_CTR_CMPC	EPWM8_CTR_CMPC	Disable
59	EPWM8_CTR_CMPD	EPWM8_CTR_CMPD	Disable
60	EPWM8A_AQ	EPWM8A_AQ	Disable
61	EPWM8B_AQ	EPWM8B_AQ	Disable
62	EPWM8A_DB	EPWM8A_DB	Enable
63	EPWM8B_DB	EPWM8B_DB	Enable
64	AUXSIG0	AUXSIG0	Enable
65	AUXSIG1	AUXSIG1	Enable
66	AUXSIG2	AUXSIG2	Enable
67	AUXSIG3	AUXSIG3	Enable
68	AUXSIG4	AUXSIG4	Enable
69	AUXSIG5	AUXSIG5	Enable
70	AUXSIG6	AUXSIG6	Enable
71	AUXSIG7	AUXSIG7	Enable
72	CLB5_OUT16	CLB5_OUT16	Disable
73	CLB5_OUT17	CLB5_OUT17	Disable
74	CLB5_OUT18	CLB5_OUT18	Disable
75	CLB5_OUT19	CLB5_OUT19	Disable
76	CLB5_OUT20	CLB5_OUT20	Disable
77	CLB5_OUT21	CLB5_OUT21	Disable
78	CLB5_OUT22	CLB5_OUT22	Disable
79	CLB5_OUT23	CLB5_OUT23	Disable
80	CLB6_OUT16	CLB6_OUT16	Disable
81	CLB6_OUT17	CLB6_OUT17	Disable
82	CLB6_OUT18	CLB6_OUT18	Disable
83	CLB6_OUT19	CLB6_OUT19	Disable
84	CLB6_OUT20	CLB6_OUT20	Disable
85	CLB6_OUT21	CLB6_OUT21	Disable
86	CLB6_OUT22	CLB6_OUT22	Disable
87	CLB6_OUT23	CLB6_OUT23	Disable
88	CLB3_OUT16	CLB3_OUT16	Disable
89	CLB3_OUT17	CLB3_OUT17	Disable
90	CLB3_OUT18	CLB3_OUT18	Disable
91	CLB3_OUT19	CLB3_OUT19	Disable
92	CLB3_OUT20	CLB3_OUT20	Disable

Table 11-3. Global Signals and Mux Selection (continued)

Select Value	CLB5 Input	CLB6 Input	Synchronization Requirement
93	CLB3_OUT21	CLB3_OUT21	Disable
94	CLB3_OUT22	CLB3_OUT22	Disable
95	CLB3_OUT23	CLB3_OUT23	Disable
96	CLB4_OUT16	CLB4_OUT16	Disable
97	CLB4_OUT17	CLB4_OUT17	Disable
98	CLB4_OUT18	CLB4_OUT18	Disable
99	CLB4_OUT19	CLB4_OUT19	Disable
100	CLB4_OUT20	CLB4_OUT20	Disable
101	CLB4_OUT21	CLB4_OUT21	Disable
102	CLB4_OUT22	CLB4_OUT22	Disable
103	CLB4_OUT23	CLB4_OUT23	Disable
104	CPU1_ERAD_EBC_EVT0	CPU1_ERAD_SEC_EVT0	Disable
105	CPU1_ERAD_EBC_EVT1	CPU1_ERAD_SEC_EVT1	Disable
106	CPU1_ERAD_EBC_EVT2	CPU1_ERAD_SEC_EVT2	Disable
107	CPU1_ERAD_EBC_EVT3	CPU1_ERAD_SEC_EVT3	Disable
108	CPU1_ERAD_EBC_EVT4	CPU3_ERAD_SEC_EVT0	Disable
109	CPU1_ERAD_EBC_EVT5	CPU3_ERAD_SEC_EVT1	Disable
110	CPU1_ERAD_EBC_EVT6	CPU3_ERAD_SEC_EVT2	Disable
111	CPU1_ERAD_EBC_EVT7	CPU3_ERAD_SEC_EVT3	Disable
112	FSIRXA_PING_TAG_MATCH	FSIRXA_PING_TAG_MATCH	Disable
113	FSIRXA_DATA_TAG_MATCH	FSIRXA_DATA_TAG_MATCH	Disable
114	FSIRXA_ERROR_TAG_MATCH	FSIRXA_ERROR_TAG_MATCH	Disable
115	FSIRXB_PING_TAG_MATCH	FSIRXB_PING_TAG_MATCH	Disable
116	FSIRXB_DATA_TAG_MATCH	FSIRXB_DATA_TAG_MATCH	Disable
117	FSIRXB_ERROR_TAG_MATCH	FSIRXB_ERROR_TAG_MATCH	Disable
118	ECAT_SOF	ECAT_SOF	Enable
119	ECAT_EOF	ECAT_EOF	Enable
120	SPIC_CLK_OUT	SPIC_CLK_OUT	Enable
121	SPIC_POCI_IN	SPIC_POCI_IN	Enable
122	SPIC_PTE_OUT	SPIC_PTE_OUT	Enable
123	SPID_CLK_OUT	SPID_CLK_OUT	Enable
124	SPID_POCI_IN	SPID_POCI_IN	Enable
125	SPID_PTE_OUT	SPID_PTE_OUT	Enable
126	ECAT_SYNC0	ECAT_SYNC0	Enable
127	ECAT_SYNC1	ECAT_SYNC1	Enable

Note

EPWMxA_OE and EPWMxB_OE refer to trip outputs from the respective EPWM module.

EPWMxA_AQ and EPWMxB_AQ refer to the output of the AQ submodule in the respective EPWM module.

EPWMxA_DB and EPWMBx_DB refer to the output of the DB submodule in the respective EPWM module.

Note

If a signal in the following table indicates that synchronization is required, then the CLB input synchronizer must be enabled using the appropriate SYNC bit in the CLB_INPUT_FILTER register. This synchronization adds a 2-3 CLB clock cycle delay to the input. This delay is either 2 or 3 cycles and is not predictable. There is a potential for a metastability hazard, if the indicated signals are not first synchronized before going into the CLB tile. This metastability can cause errors dependent on voltage, temperature, and wafer fab process. Note that this requirement is in addition to and separate from GPIO input synchronization.

If a signal in the following table indicates that synchronization is not required, as the signal is already synchronous, then pipelining is required and must be enabled using the PIPE bit in the CLB_INPUT_FILTER register. This pipelining adds a 1 CLB clock cycle delay to the input. This is not to be mistaken with the PIPELINE_EN bit in the CLB_LOAD_EN register, which controls pipelining of the CLB operations in the HLC and counter blocks. Having synchronization and pipelining both enabled or both disabled is not recommended. Enabling both synchronization and pipelining introduces a delay of more than 2-3 CLB clock cycles on the signal path. Disabling both allows the completely asynchronous signal to be routed as an input.

Table 11-4. Local Signals and Mux Selection

Select Value	CLB1 Input	CLB2 Input	CLB3 Input	CLB4 Input	Synchronization Requirement
0	CLB1_GLB_MUX_OUT	CLB2_GLB_MUX_OUT	CLB3_GLB_MUX_OUT	CLB4_GLB_MUX_OUT	Enable
1	EPWM1_DCAEVT1	EPWM2_DCAEVT1	EPWM3_DCAEVT1	EPWM4_DCAEVT1	Enable
2	EPWM1_DCAEVT2	EPWM2_DCAEVT2	EPWM3_DCAEVT2	EPWM4_DCAEVT2	Enable
3	EPWM1_DCBEVT1	EPWM2_DCBEVT1	EPWM3_DCBEVT1	EPWM4_DCBEVT1	Enable
4	EPWM1_DCBEVT2	EPWM2_DCBEVT2	EPWM3_DCBEVT2	EPWM4_DCBEVT2	Enable
5	EPWM1_DCAH	EPWM2_DCAH	EPWM3_DCAH	EPWM4_DCAH	Enable
6	EPWM1_DCAL	EPWM2_DCAL	EPWM3_DCAL	EPWM4_DCAL	Enable
7	EPWM1_DCBH	EPWM2_DCBH	EPWM3_DCBH	EPWM4_DCBH	Enable
8	EPWM1_DCBL	EPWM2_DCBL	EPWM3_DCBL	EPWM4_DCBL	Enable
9	EPWM1_OST	EPWM2_OST	EPWM3_OST	EPWM4_OST	Enable
10	EPWM1_CBC	EPWM2_CBC	EPWM3_CBC	EPWM4_CBC	Enable
11	ECAP1IN0	ECAP2IN0	ECAP3IN0	ECAP4IN0	Enable
12	ECAP1_OUT	ECAP2_OUT	ECAP3_OUT	ECAP4_OUT	Disable
13	ECAP1_OUT_EN	ECAP2_OUT_EN	ECAP3_OUT_EN	ECAP4_OUT_EN	Disable
14	ECAP1_CEVT1	ECAP2_CEVT1	ECAP3_CEVT1	ECAP4_CEVT1	Disable
15	ECAP1_CEVT2	ECAP2_CEVT2	ECAP3_CEVT2	ECAP4_CEVT2	Disable
16	ECAP1_CEVT3	ECAP2_CEVT3	ECAP3_CEVT3	ECAP4_CEVT3	Disable

Table 11-4. Local Signals and Mux Selection (continued)

Select Value	CLB1 Input	CLB2 Input	CLB3 Input	CLB4 Input	Synchronization Requirement
17	ECAP1_CEVT4	ECAP2_CEVT4	ECAP3_CEVT4	ECAP4_CEVT4	Disable
18	EQEP1A	EQEP2A	EQEP3A	EQEP4A	Enable
19	EQEP1B	EQEP2B	EQEP3B	EQEP4B	Enable
20	EQEP1I	EQEP2I	EQEP3I	EQEP4I	Enable
21	EQEP1S	EQEP2S	EQEP3S	EQEP4S	Enable
22	CPU1_TBCLKSYNC	CPU1_TBCLKSYNC	CPU1_TBCLKSYNC	CPU1_TBCLKSYNC	Enable
23	CPU3_TBCLKSYNC	CPU3_TBCLKSYNC	CPU3_TBCLKSYNC	CPU3_TBCLKSYNC	Enable
24	CPU1_HALT	CPU1_HALT	CPU1_HALT	CPU1_HALT	Enable
25	SPIA_PICO_OUT	SPIB_PICO_OUT	SPIC_PICO_OUT	SPID_PICO_OUT	Enable
26	SPIA_CLK_IN	SPIB_CLK_IN	SPIC_CLK_IN	SPID_CLK_IN	Enable
27	SPIA_PICO_IN	SPIB_PICO_IN	SPIC_PICO_IN	SPID_PICO_IN	Enable
28	SPIA_PTE_IN	SPIB_PTE_IN	SPIC_PTE_IN	SPID_PTE_IN	Enable
29	Reserved	Reserved	Reserved	Reserved	Reserved
30	SPIA_POCI_OUT	SPIB_POCI_OUT	SPIC_POCI_OUT	SPID_POCI_OUT	Enable
31	CLB1_PSCLK	CLB2_PSCLK	CLB3_PSCLK	CLB4_PSCLK	Enable
32	EPWM9A	EPWM9A	EPWM9A	EPWM9A	Enable
33	EPWM9A_OE	EPWM9A_OE	EPWM9A_OE	EPWM9A_OE	Enable
34	EPWM9B	EPWM9B	EPWM9B	EPWM9B	Enable
35	EPWM9B_OE	EPWM9B_OE	EPWM9B_OE	EPWM9B_OE	Enable
36	EPWM10A	EPWM10A	EPWM10A	EPWM10A	Enable
37	EPWM10A_OE	EPWM10A_OE	EPWM10A_OE	EPWM10A_OE	Enable
38	EPWM10B	EPWM10B	EPWM10B	EPWM10B	Enable
39	EPWM10B_OE	EPWM10B_OE	EPWM10B_OE	EPWM10B_OE	Enable
40	EPWM11A	EPWM11A	EPWM11A	EPWM11A	Enable
41	EPWM11A_OE	EPWM11A_OE	EPWM11A_OE	EPWM11A_OE	Enable
42	EPWM11B	EPWM11B	EPWM11B	EPWM11B	Enable
43	EPWM11B_OE	EPWM11B_OE	EPWM11B_OE	EPWM11B_OE	Enable
44	EPWM12A	EPWM12A	EPWM12A	EPWM12A	Enable
45	EPWM12A_OE	EPWM12A_OE	EPWM12A_OE	EPWM12A_OE	Enable
46	EPWM12B	EPWM12B	EPWM12B	EPWM12B	Enable
47	EPWM12B_OE	EPWM12B_OE	EPWM12B_OE	EPWM12B_OE	Enable
48	INPUTXBAR17	INPUTXBAR17	INPUTXBAR17	INPUTXBAR17	Enable
49	INPUTXBAR18	INPUTXBAR18	INPUTXBAR18	INPUTXBAR18	Enable
50	INPUTXBAR19	INPUTXBAR19	INPUTXBAR19	INPUTXBAR19	Enable
51	INPUTXBAR20	INPUTXBAR20	INPUTXBAR20	INPUTXBAR20	Enable
52	INPUTXBAR21	INPUTXBAR21	INPUTXBAR21	INPUTXBAR21	Enable
53	INPUTXBAR22	INPUTXBAR22	INPUTXBAR22	INPUTXBAR22	Enable

Table 11-4. Local Signals and Mux Selection (continued)

Select Value	CLB1 Input	CLB2 Input	CLB3 Input	CLB4 Input	Synchronization Requirement
54	INPUTXBAR23	INPUTXBAR23	INPUTXBAR23	INPUTXBAR23	Enable
55	INPUTXBAR24	INPUTXBAR24	INPUTXBAR24	INPUTXBAR24	Enable
56	INPUTXBAR25	INPUTXBAR25	INPUTXBAR25	INPUTXBAR25	Enable
57	INPUTXBAR26	INPUTXBAR26	INPUTXBAR26	INPUTXBAR26	Enable
58	INPUTXBAR27	INPUTXBAR27	INPUTXBAR27	INPUTXBAR27	Enable
59	INPUTXBAR28	INPUTXBAR28	INPUTXBAR28	INPUTXBAR28	Enable
60	INPUTXBAR29	INPUTXBAR29	INPUTXBAR29	INPUTXBAR29	Enable
61	INPUTXBAR30	INPUTXBAR30	INPUTXBAR30	INPUTXBAR30	Enable
62	INPUTXBAR31	INPUTXBAR31	INPUTXBAR31	INPUTXBAR31	Enable
63	INPUTXBAR32	INPUTXBAR32	INPUTXBAR32	INPUTXBAR32	Enable

Table 11-5. Local Signals and Mux Selection

Select Value	CLB5 Input	CLB6 Input	Synchronization Requirement
0	CLB5_GLB_MUX_OUT	CLB6_GLB_MUX_OUT	Enable
1	EPWM5_DCAEVT1	EPWM6_DCAEVT1	Enable
2	EPWM5_DCAEVT2	EPWM6_DCAEVT2	Enable
3	EPWM5_DCBEVT1	EPWM6_DCBEVT1	Enable
4	EPWM5_DCBEVT2	EPWM6_DCBEVT2	Enable
5	EPWM5_DCAH	EPWM6_DCAH	Enable
6	EPWM5_DCAL	EPWM6_DCAL	Enable
7	EPWM5_DCBH	EPWM6_DCBH	Enable
8	EPWM5_DCBL	EPWM6_DCBL	Enable
9	EPWM5_OST	EPWM6_OST	Enable
10	EPWM5_CBC	EPWM6_CBC	Enable
11	ECAP3IN0	ECAP4IN0	Enable
12	ECAP3_OUT	ECAP4_OUT	Disable
13	ECAP3_OUT_EN	ECAP4_OUT_EN	Disable
14	ECAP3_CEVT1	ECAP4_CEVT1	Disable
15	ECAP3_CEVT2	ECAP4_CEVT2	Disable
16	ECAP3_CEVT3	ECAP4_CEVT3	Disable
17	ECAP3_CEVT4	ECAP4_CEVT4	Disable
18	FSIRXC_DATA_PKT_RCVD	FSIRXD_DATA_PKT_RCVD	Disable
19	FSIRXC_ERROR_PKT_RCVD	FSIRXD_ERROR_PKT_RCVD	Disable
20	FSIRXC_PING_PKT_RCVD	FSIRXD_PING_PKT_RCVD	Disable
21	CPU3_HALT	CPU3_HALT	Disable
22	CPU1_TBCLKSYNC	CPU1_TBCLKSYNC	Enable
23	CPU2_TBCLKSYNC	CPU2_TBCLKSYNC	Enable

Table 11-5. Local Signals and Mux Selection (continued)

Select Value	CLB5 Input	CLB6 Input	Synchronization Requirement
24	CPU1_HALT	CPU1_HALT	Enable
25	SPIC_PICO_OUT	SPID_PICO_OUT	Enable
26	SPIC_CLK_IN	SPID_CLK_IN	Enable
27	SPIC_PICO_IN	SPID_PICO_IN	Enable
28	SPIC_PTE_IN	SPID_PTE_IN	Enable
29	Reserved	Reserved	Reserved
30	SPIC_POCI_OUT	SPID_POCI_OUT	Enable
31	CLB5_PSCLK	CLB6_PSCLK	Enable
32	ECAP5IN0	ECAP6IN0	Enable
33	ECAP5_OUT	ECAP6_OUT	Disable
34	ECAP5_OUT_EN	ECAP6_OUT_EN	Disable
35	ECAP5_CEVT1	ECAP6_CEVT1	Disable
36	ECAP5_CEVT2	ECAP6_CEVT2	Disable
37	ECAP5_CEVT3	ECAP6_CEVT3	Disable
38	ECAP5_CEVT4	ECAP6_CEVT4	Disable
39	ECAP5IN0	ECAP6IN0	Enable
40	EQEP5A	EQEP6A	Enable
41	EQEP5B	EQEP6B	Enable
42	EQEP5I	EQEP6I	Enable
43	EQEP5S	EQEP6S	Enable
44	EPWM16A	EPWM16A	Enable
45	EPWM16A_OE	EPWM16A_OE	Enable
46	EPWM16B	EPWM16B	Enable
47	EPWM16B_OE	EPWM16B_OE	Enable
48	INPUTXBAR17	INPUTXBAR17	Enable
49	INPUTXBAR18	INPUTXBAR18	Enable
50	INPUTXBAR19	INPUTXBAR19	Enable
51	INPUTXBAR20	INPUTXBAR20	Enable
52	INPUTXBAR21	INPUTXBAR21	Enable
53	INPUTXBAR22	INPUTXBAR22	Enable
54	INPUTXBAR23	INPUTXBAR23	Enable
55	INPUTXBAR24	INPUTXBAR24	Enable
56	INPUTXBAR25	INPUTXBAR25	Enable
57	INPUTXBAR26	INPUTXBAR26	Enable
58	INPUTXBAR27	INPUTXBAR27	Enable
59	INPUTXBAR28	INPUTXBAR28	Enable
60	INPUTXBAR29	INPUTXBAR29	Enable
61	INPUTXBAR30	INPUTXBAR30	Enable

Table 11-5. Local Signals and Mux Selection (continued)

Select Value	CLB5 Input	CLB6 Input	Synchronization Requirement
62	INPUTXBAR31	INPUTXBAR31	Enable
63	INPUTXBAR32	INPUTXBAR32	Enable

The GPREG is accessible by the CPU and the bits of this register can be used as BOUNDARY INPUTs for the CLB Tiles. For example, CLB1s GPREG[0] can be used as BOUNDARY IN0 (Cell Input 0) for the corresponding CLB Tile.

To connect multiple tiles to each other, you can use the CLBx OUT4/5 and connect to CLBy BOUNDARY INz through the CLB X-BAR and the Global Signals Mux.

Another option is to connect the CLBx OUT0-7 to a GPIO and then use the INPUT X-BAR to bring the signal back in to the device and connect to the CLBy BOUNDARY INz through the CLB X-BAR and the Global Signals Mux.

To use GPIOs as inputs to the CLB, you must utilize the Input X-BAR and the CLB X-BAR. Figure 11-4 shows how GPIOs can be used as inputs to the CLB tiles.

11.3.3 CLB Output Selection

The eight outputs of the CLB are replicated to create 32 output signals. Each of these outputs has a separate enable bit defined in the CLB output enable register, CLB_OUT_EN. The CLB outputs go to the ePWM, eCAP, eQEP and the crossbar module in the device. This allows the user to enhance the functionality of these modules with the CLB. Figure 11-8 shows the CLB outputs.

The user has the capability to disable updated to the CLB_OUT_EN register by blocking access to the register through setting the CLB_MISC_ACCESS_CTRL.BLKEN bit. The eight outputs are replicated to generate a total of 32 outputs (shown in Figure 11-8). Some of these new outputs can be used for TILE to TILE connection through the CLB Global Mux inputs.

Note

The output from OUTLUT0 is connected to OUT0, OUT8, OUT16, and OUT24. While the signal is the same, each OUTy has access to a different peripheral as shown in Section 11.3.4. Likewise, OUTLUT1 is connected to OUT1, OUT9, OUT17, and OUT25, and are the same signal.

CLBx_OUT12 through CLBx_OUT15 are unregistered and asynchronous to the CLB clock.

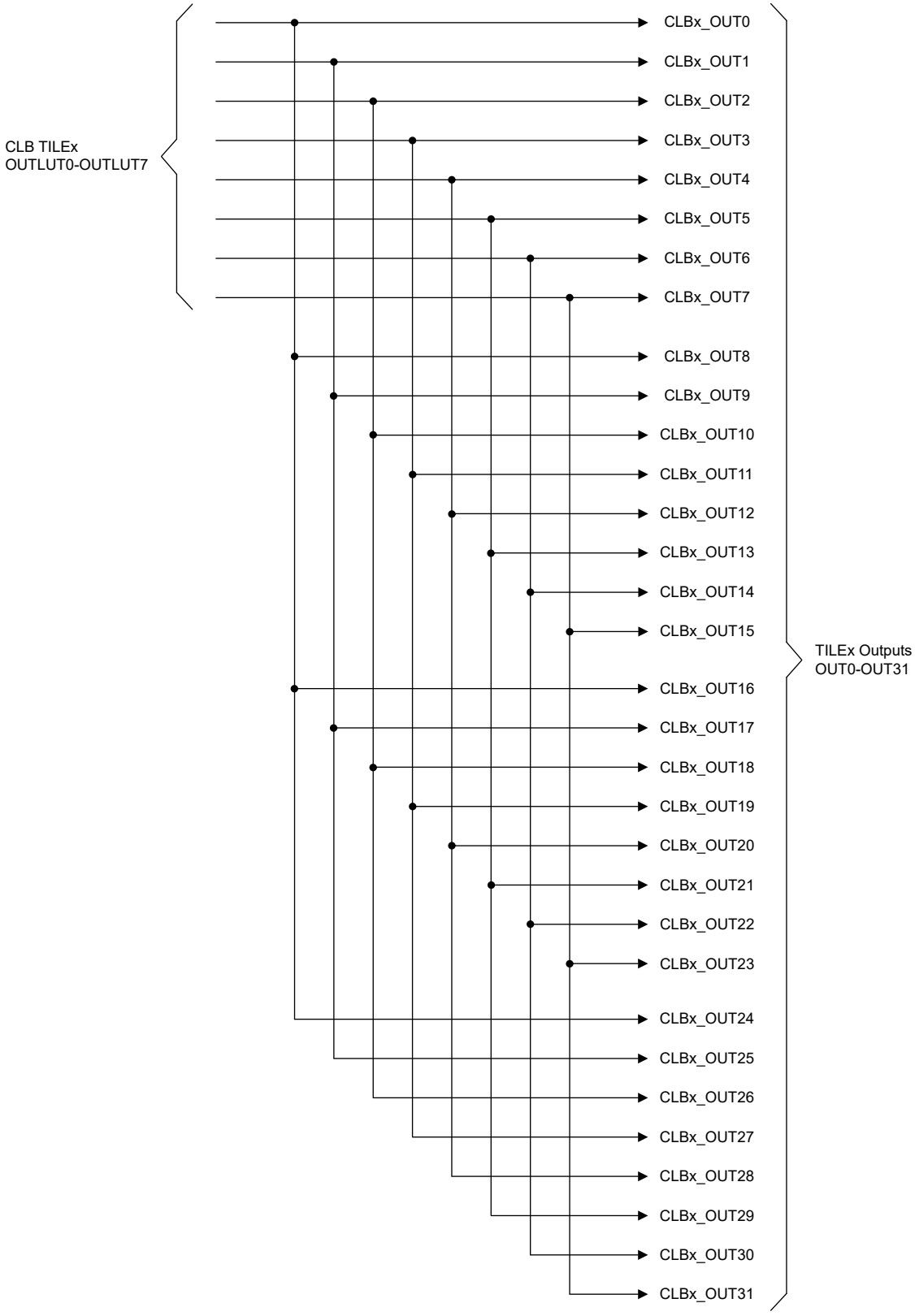


Figure 11-8. CLB Outputs

11.3.4 CLB Output Signal Multiplexer

Each CLB output signal passes through an external multiplexer that intersects a specific peripheral signal, see Figure 11-9. The output of the multiplexer is connected to the destination of the original peripheral signal and the default multiplexer setting is that the peripheral signal is passed through. The multiplexer is controlled by bit[n] in the CLB output enable register CLB_OUT_EN.

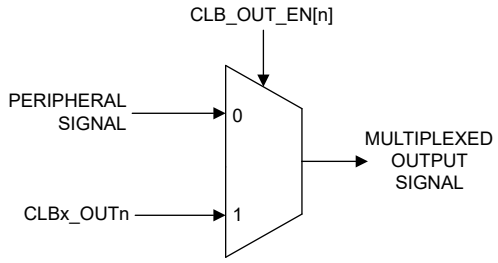


Figure 11-9. CLB Output Signal Multiplexer

For example, if the CLB1 OUT0 must override the EPWM1A signal, the OUPTUT ENABLE bit for OUT0 must be set to 1.

Table 11-6 and Table 11-7 shows the allocation of peripheral signals and the CLB outputs.

Table 11-6. CLB Output Signal Multiplexer Table

CLB Output	CLB1 Destination	CLB2 Destination	CLB3 Destination	CLB4 Destination
0	HRPWM1A	HRPWM2A	HRPWM3A	HRPWM4A
1	HRPWM1A_OE	HRPWM2A_OE	HRPWM3A_OE	HRPWM4A_OE
2	HRPWM1B	HRPWM2B	HRPWM3B	HRPWM4B
3	HRPWM1B_OE	HRPWM2B_OE	HRPWM3B_OE	HRPWM4B_OE
4	EPWM1A_AQ	EPWM2A_AQ	EPWM3A_AQ	EPWM4A_AQ
5	EPWM1B_AQ	EPWM2B_AQ	EPWM3B_AQ	EPWM4B_AQ
6	EPWM1A_DB	EPWM2A_DB	EPWM3A_DB	EPWM4A_DB
7	EPWM1B_DB	EPWM2B_DB	EPWM3B_DB	EPWM4B_DB
8	EQEP1_QCLK	EQEP2_QCLK	EQEP3_QA	EQEP4_QA
9	EQEP1_QDIR	EQEP2_QDIR	EQEP3_QB	EQEP4_QB
10	EQEP1_QB	EQEP2_QB	EQEP3_QDIR	EQEP4_QDIR
11	EQEP1_QA	EQEP2_QA	EQEP3_QCLK	EQEP4_QCLK
12	All XBARs	All XBARs	All XBARs	All XBARs
13	All XBARs	All XBARs	All XBARs	All XBARs
14	ECAP Mux	ECAP Mux	ECAP Mux	ECAP Mux
15	ECAP Mux	ECAP Mux	ECAP Mux	ECAP Mux
16	Global Mux	Global Mux	Global Mux	Global Mux
17	Global Mux	Global Mux	Global Mux	Global Mux
18	Global Mux	Global Mux	Global Mux	Global Mux
19	Global Mux	Global Mux	Global Mux	Global Mux
20	Global Mux	Global Mux	Global Mux	Global Mux
21	Global Mux, SPIA_PTE_OUT	Global Mux, SPIB_PTE_OUT	Global Mux, SPIC_PTE_OUT	Global Mux, SPIPTE_OUT

Table 11-6. CLB Output Signal Multiplexer Table (continued)

CLB Output	CLB1 Destination	CLB2 Destination	CLB3 Destination	CLB4 Destination
22	Global Mux, SPIA_PICO_OUT	Global Mux, SPIB_PICO_OUT	Global Mux, SPIC_PICO_OUT	Global Mux, SPID_PICO_OUT
23	Global Mux, SPIA_POCI_OUT	Global Mux, SPIB_POCI_OUT	Global Mux, SPIC_POCI_OUT	Global Mux, SPID_POCI_OUT
24	SPIA_CLK_IN	SPIB_CLK_IN	SPIC_CLK_IN	SPID_CLK_IN
25	SPIA_PICO_IN	SPIB_PICO_IN	SPIC_PICO_IN	SPID_PICO_IN
26	SPIA_PTE_IN	SPIB_PTE_IN	SPIC_PTE_IN	SPID_PTE_IN
27	ADC_SOC_TRIGGERS	ADC_SOC_TRIGGERS	ADC_SOC_TRIGGERS	ADC_SOC_TRIGGERS
28	ECAP1_OUT_EN	ECAP2_OUT_EN	ECAP3_OUT_EN	ECAP4_OUT_EN
29	ECAP1_OUT	ECAP2_OUT	ECAP3_OUT	ECAP4_OUT
30	FSITX Triggers	FSITX Triggers	FSITX Triggers	FSITX Triggers
31	FSITX Triggers	FSITX Triggers	FSITX Triggers	FSITX Triggers

Table 11-7. CLB Output Signal Multiplexer Table

CLB Output	CLB5 Destination	CLB6 Destination
0	HRPWM5A	HRPWM6A
1	HRPWM5A_OE	HRPWM6A_OE
2	HRPWM5B	HRPWM6B
3	HRPWM5B_OE	HRPWM6B_OE
4	EPWM5A_AQ	EPWM6A_AQ
5	EPWM5B_AQ	EPWM6B_AQ
6	EPWM5A_DB	EPWM6A_DB
7	EPWM5B_DB	EPWM6B_DB
8	EQEP5_QA	EQEP6_QA
9	EQEP5_QB	EQEP6_QB
10	EQEP5_QDIR	EQEP6_QDIR
11	EQEP5_QCLK	EQEP6_QCLK
12	All XBARs	All XBARs
13	All XBARs	All XBARs
14	ECAP Mux	ECAP Mux
15	ECAP Mux	ECAP Mux
16	Global Mux	Global Mux
17	Global Mux	Global Mux
18	Global Mux	Global Mux
19	Global Mux	Global Mux
20	Global Mux	Global Mux
21	Global Mux, SPIE_PTE_OUT	Global Mux
22	Global Mux, SPIE_PICO_OUT	Global Mux
23	Global Mux, SPIE_POCI_OUT	Global Mux

Table 11-7. CLB Output Signal Multiplexer Table (continued)

CLB Output	CLB5 Destination	CLB6 Destination
24	SPIE_CLK_IN	Reserved
25	SPIE_PICO_IN	Reserved
26	SPIE_PTE_IN	Reserved
27	ADC_SOC_TRIGGERS	ADC_SOC_TRIGGERS
28	ECAP5_OUT_EN	ECAP6_OUT_EN
29	ECAP5_OUT	ECAP6_OUT
30	FSITX Triggers	FSITX Triggers
31	FSITX Triggers	FSITX Triggers

Note

When not explicitly specified in the table above, refer to corresponding IP chapters for exact CLB Output destinations (that is, FSI, ECAP, XBAR connections).

11.4 CLB Tile

The purpose of the CLB tile is to provide the logic reconfiguration capability of the CLB. The CLB tile contains the following submodules:

- **Counter:** The counter submodule can be configured as an adder, a counter, or a shifter. When functioning as an adder, the counter submodule can either add or subtract. When functioning as a counter, the counter submodule can count up or count down. When functioning as a shifter, the counter submodule can shift left or shift right. The counter event inputs, as well as the reset input, can be freely connected to any of the other submodules in the same tile. Starting with CLB Type 2, the counter module can also operate as a serializer or linear feedback shift register. There are three counters in each tile.
- **LUT4:** The LUT4 submodule has a 4-input look-up table functionality and is capable of realizing any combinatorial Boolean equation of up to four inputs. There are three LUT4 submodules in each CLB tile.
- **FSM:** The Finite State Machine (FSM) submodule can be configured either as a single four-state finite state machine, or as two independent two-state finite state machines. The FSM accepts two external inputs, and generates two state outputs and one combination output. When not used as a state machine, the FSM submodule can accept two external inputs and function as a 4-input LUT. There are three FSM submodules in each CLB tile.
- **Output LUT:** The output LUT is a 3-input lookup table submodule capable of realizing any combinatorial Boolean equation of up to three inputs. There are eight such blocks in a CLB tile, each associated with one of the tile outputs.
- **Asynchronous Output Conditioning Block:** The primary purpose of the Asynchronous Output Conditioning (AOC) block is to provide asynchronous conditioning capabilities on the TILE outputs or directly on the inputs of the TILE.
- **High Level Controller:** The High Level Controller (HLC) submodule is an event-driven block that can handle up to four concurrent events. The event can be an activity on any of the other block outputs. A predefined set of operations is executed when each event occurs. The HLC also provides a data exchange and interrupt mechanism to the CPU subsystem. There are four working registers (R0, R1, R2, and R3) that can be used for basic operations, and to modify or set up values for the three counter blocks. Unlike the other submodules, there is only one HLC in each CLB tile.
- **Static Switch Block:** The static switch block provides dynamic connectivity between all of the blocks listed above. Submodules can be connected by the user, with the only restriction that the submodules must not form a combinational loop within the tile.

A CLB tile consists of three sets each of the counter block, FSM, and LUT4, one high-level controller, and eight output LUT blocks. The submodule numbering is shown in Figure 11-10.

The functionality of the LUT submodules is configured using a register field containing the binary pattern of the output of the desired look-up table. For example, a 4-input LUT has 16 possible input permutations, each of which corresponds to a desired binary 0 or 1 at the output. The register field can, therefore, be 16-bits in length, with each bit representing the desired result of a binary pattern. Input pattern sequences start at 0000 and continue sequentially to 1111. A similar method is used to encode the 16-bit state equations in the FSM submodule.

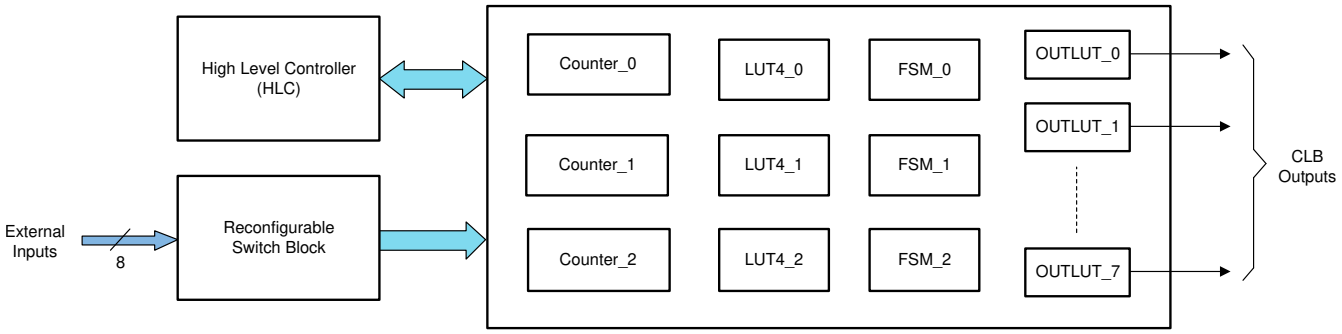


Figure 11-10. CLB Tile Submodules

11.4.1 Static Switch Block

The Static Switch Block provides the configurable connectivity between the submodules in the CLB tile. The outputs of all the submodules and the eight external inputs are connected to a common internal bus inside the tile. Every input port has a 32-to-1 multiplexer and an associated 5-bit selection value that allows the user to select one of the inputs on the bus. The only restrictions are certain signals (described below) that are tied off in the design to prevent creation of accidental combinatorial loops.

Table 11-8. Output Table

Bit Position	Signal Connection	Comment
0	Always 0	This select value is used to tie an input to 0.
1	COUNTER_0 MATCH2	
2	COUNTER_0 ZERO	
3	COUNTER_0 MATCH1	
4	FSM_0 STATE_BIT_0	
5	FSM_0 STATE_BIT_1	
6	FSM_0 LUT output	
7	LUT4_0 output	
8	Always 1	This select value is used to tie an input to 1.
9	COUNTER_1 MATCH2	
10	COUNTER_1 ZERO	
11	COUNTER_1 MATCH1	
12	FSM_1 STATE_BIT_0	
13	FSM_1 STATE_BIT_1	
14	FSM_1 LUT output	
15	LUT4_1 output	
16	Always ‘0’	
17	COUNTER_2 MATCH2	
18	COUNTER_2 ZERO	
19	COUNTER_2 MATCH1	
20	FSM_2 STATE_BIT_0	
21	FSM_2 STATE_BIT_1	
22	FSM_2 LUT output	
23	LUT4_2 output	
24	External Input 0	
25	External Input 1	
26	External Input 2	
27	External Input 3	
28	External Input 4	
29	External Input 5	
30	External Input 6	
31	External Input 7	

Table 11-9. Input Table

Module Name	Port Name	Description
Counter Block	RESET	Acts as an active high reset when used as a counter
	MODE_0	Acts as an enable when used as a counter. The counter counts only when this input is 1.
	MODE_1	Acts as a direction control when used as a counter. If this input is 1, then the counter counts up; else, the counter counts down.
LUT	IN0	Input 0 of the 4-input LUT.
	IN1	Input 1 of the 4-input LUT.
	IN2	Input 2 of the 4-input LUT.
	IN3	Input 3 of the 4-input LUT.
FSM	EXT_IN0	Input 0 of the FSM block.
	EXT_IN1	Input 1 of the FSM block.
	EXTRA_EXT_IN0	Extra external input 0 of the FSM block. This input matters only if configured in the LUT mode.
	EXTRA_EXT_IN1	Extra external input 1 of the FSM block. This input matters only if configured in the LUT mode.

The static switch block allows the user to define the input connection of any submodule to come from any of the outputs in Table 11-8. It is therefore easy to create a combinatorial loop. To prevent this, certain paths are broken in the input path of each submodule. These port positions are tied to 0, as shown in Table 11-10.

Table 11-10. Ports Tied Off to Prevent Combinatorial Loops

Module Name	Ports of Input MUX Tied Off to 0 to Prevent Combinational Loops
LUT_0	LUT_0 , LUT_1, and LUT_2 output, FSM_0, FSM_1, and FSM_2 output.
FSM_0	LUT_1 and LUT_2 output, FSM_0, FSM_1, and FSM_2 output.
LUT_1	LUT_1 and LUT_2 output, FSM_1 and FSM_2 output.
FSM_1	LUT_2 output, FSM_1 and FSM_2 output.
LUT_2	LUT_2 output, FSM_2 output.
FSM_2	FSM_2 output.

11.4.2 Counter Block

11.4.2.1 Counter Description

The counter block is a complex functional submodule that can be configured either as a counter, an adder, or a shifter. Apart from the normal operational control, this block has a dedicated EVENT input, which can trigger an addition, subtraction or shift operation, or load data into the counter register. The inputs to the counter submodule are shown in Figure 11-11.

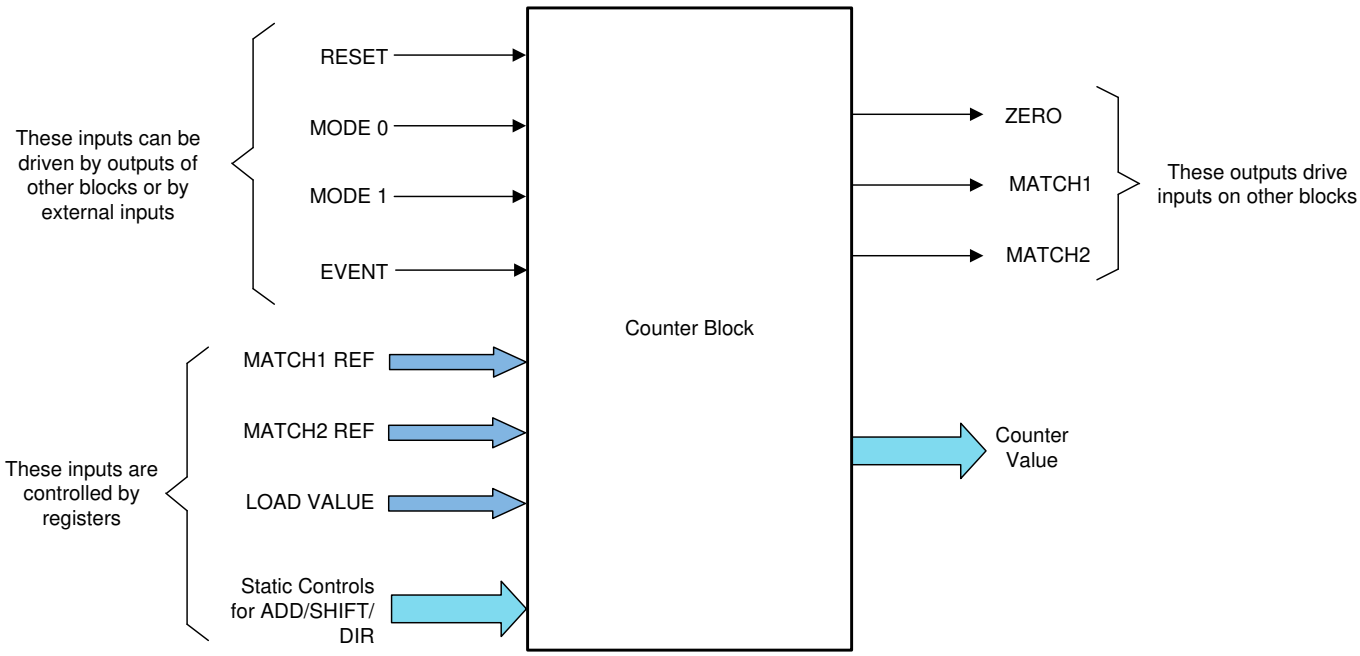


Figure 11-11. Counter Block

11.4.2.2 Counter Operation

At the heart of the counter block is a 32-bit count register. This register can either be loaded statically before counting commences, or dynamically at run time. The operation of the counter submodule is determined by the inputs described below. Note that each of the inputs can be connected to the outputs of any of the other blocks in the CLB tile. These connections are made by configuring the static switch block.

The counter inputs are:

- **RESET:** This is the highest priority input and takes precedence over all other inputs. The input is level sensitive and as long as the input remains high, the counter resets to 0 on the next clock cycle.
- **MODE_0:** This input is an enable for the counter. The counter begins counting (up or down depending on the MODE_1 setting) only when this input is high. If this input is low, then no counting takes place.
- **MODE_1:** This input is the direction control for the counter. If this input is high, then the counter increments on every clock cycle where MODE_0 is high. If this input is low, then the counter decrements on every clock cycle where MODE_0 is high. The counter wraps around to 0x0000 0000 after 0xFFFF FFFF when counting up. The counter wraps around to 0xFFFF FFFF after 0x0000 0000 when counting down. The only exception to this is when an EVENT occurs at exactly the same time, causing a different value to be loaded into the counter.
- **EVENT:** This input is defined for the purpose of triggering actions in the counter based on certain events. The event can be any of the outputs of the other blocks or an external input to the tile. The counter's static control inputs define the behavior of the counter on an active event. An active event is defined as a rising edge on the EVENT input. The counter can be configured to perform one of the following actions:
  - Load a predefined 32-bit value from the LOAD VALUE register into the count register
  - Shift the contents of the counter register left or right by a predefined amount between 0 and 31
  - Add or subtract a predefined 32-bit value. Addition and subtraction are treated as 32-bit unsigned operations and there is no saturation.

  Note that the effect of a rising edge on the EVENT input only lasts for one cycle. On the next cycle, the counter operation continues based on the MODE_0, MODE_1, and RESET inputs.

MATCH1 REF and MATCH2 REF are 32-bit reference values that are used to generate the MATCH1 and MATCH2 outputs. The MATCH1 output becomes active high whenever the counter register value matches the 32-bit MATCH1 REF value. MATCH2 behaves in a similar manner in relation to the MATCH2 REF register. The reference values for MATCH1 and MATCH2 can either be setup once before the start of operation, or can be modified dynamically. The High Level Controller can load desired values into the MATCH1 REF and MATCH2 REF registers.

Note that the counter load and match registers are not memory-mapped. For more information, see Section 11.5.2.

The three logic outputs of the counter block are:

- **ZERO:** This output goes high whenever the counter register is 0.
- **MATCH1:** This output goes high whenever the counter register is equal to the MATCH1 REF input register.
- **MATCH2:** This output goes high whenever the counter register is equal to the MATCH2 REF input register.

The operation of the counter block is controlled by the CFG_MISC_CTRL register. The following three bits of this register are relevant for each counter. The "x" below refers to the counter instance; 0, 1, or 2. For more information, see the CLB_MISC_CONTROL register description located in Section 11.10.

- COUNT_EVENT_CTRL_x: This bit defines whether the counter performs an addition or a shift on an event. A value of 0 means that on an event, the counter loads the static value; 1 means an add/shift operation is performed. This bit must be 0 for indirect loads and HLC loads of the counter to take effect.
- COUNT_ADD_SHIFT_x. 1 means add, 0 means shift.
- COUNT_DIR_x. 1 means left shift or add. 0 means right shift or subtract.

Table 11-11 shows the logical operation of the counter block in terms of the inputs and control register bits. Count up and down modes are the normal operation with EVENT = 0. The operations on the CNTVAL register are:

Load: CNTVAL = EVENT_LOAD_VAL

Shift right: CNTVAL = CNTVAL >> EVENT_LOAD_VAL

Shift left: CNTVAL = CNTVAL << EVENT_LOAD_VAL

Subtract: CNTVAL = CNTVAL - EVENT_LOAD_VAL

Add: CNTVAL = CNTVAL + EVENT_LOAD_VAL

Table 11-11. Counter Block Operating Modes

EVENT	MODE_0	MODE_1	COUNT_EVENT_CTRL_x	COUNT_ADD_SHIFT_x	COUNT_DIR_x	Action on CNTVAL
0	0	0	X	X	X	None
0	0	1	X	X	X	None
0	1	0	X	X	X	Count down
0	1	1	X	X	X	Count up
1	X	X	0	X	X	Load
1	X	X	1	0	0	Shift right
1	X	X	1	0	1	Shift left
1	X	X	1	1	0	Subtract
1	X	X	1	1	1	Add

11.4.2.3 Serializer Mode

Starting with CLB Type 2, the Counter module can operate as a serializer. In this mode of operation, this module acts as a shift register (also referred to as a serializer). In serializer mode of operation, the EVENT input is used to shift one bit of data into the serializer. Either of MATCH1 and MATCH2 can be configured to send out the shift register data.Using the MATCH1/2_TAP_SEL bit of CLB_COUNT_MATCH_TAP_SEL, any bit position of the counter can be brought out on the MATCH1/MATCH2 outputs. The shifting and direction of the counter in this mode is controlled by MODE_0 (enable) and MODE_1 (direction).

To enable the Serializer mode, CLB_MISC_CONTROL.COUNT_SERIALIZER_0 (for Counter 0) must be set.

11.4.2.4 Linear Feedback Shift Register (LFSR) Mode

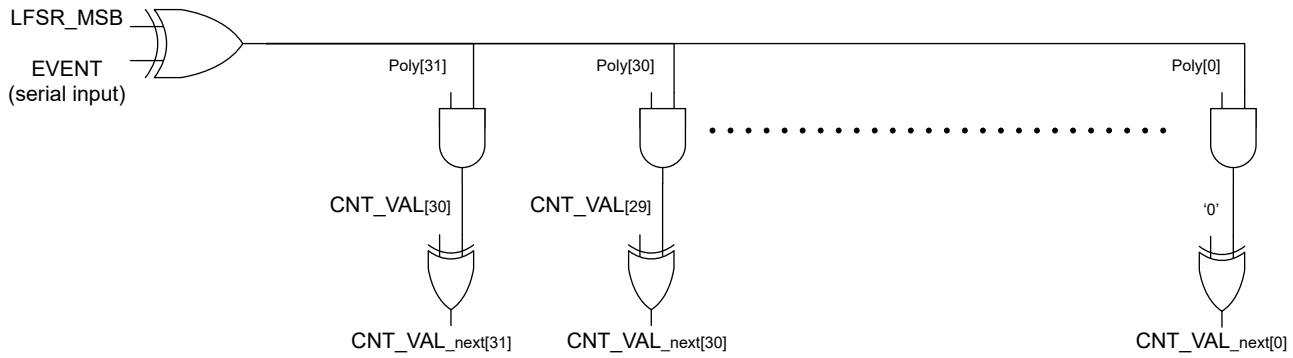
Starting with CLB Type 2, the Counter module operates as a linear feedback shift register. By configuring the characteristics of the LFSR, the counter module is used to compute the CRC on a serial bit stream. The polynomial for LFSR is in the MATCH2 reference register. The feedback bit position is in the MATCH1 reference register.

To enable the LFSR mode, CLB_MISC_CONTROL.COUNT_SERIALIZER_0 (for Counter 0) must be set along with COUNT0_LFSR_EN.

There are two types of LSFR that can be selected by changing the MODE1 value (0 or 1), as shown in Figure 11-12.

Structure for LFSR Type 1 (MODE_1 = 0)

CNT_VAL is the 32-bit counter's active register
CNT_VAL_next is the 32-bit value to be written into CNT_VAL on the next active cycle (clock edge when MODE_0 == 1)
LFSR_MSB = CNT_VAL[MATCH1_REF[4:0]]
Poly[31:0] is MATCH2_REF which acts as the CRC polynomial



Structure for LFSR Type 2 (MODE_1 = 1)

CNT_VAL is the 32-bit counter register
CNT_VAL_next is the 32-bit value to be written into CNT_VAL on the next active cycle (clock edge when MODE_0 == 1)
LFSR_MSB = CNT_VAL[MATCH1_REF[4:0]]
Poly[31:0] is MATCH2_REF which acts as the CRC polynomial

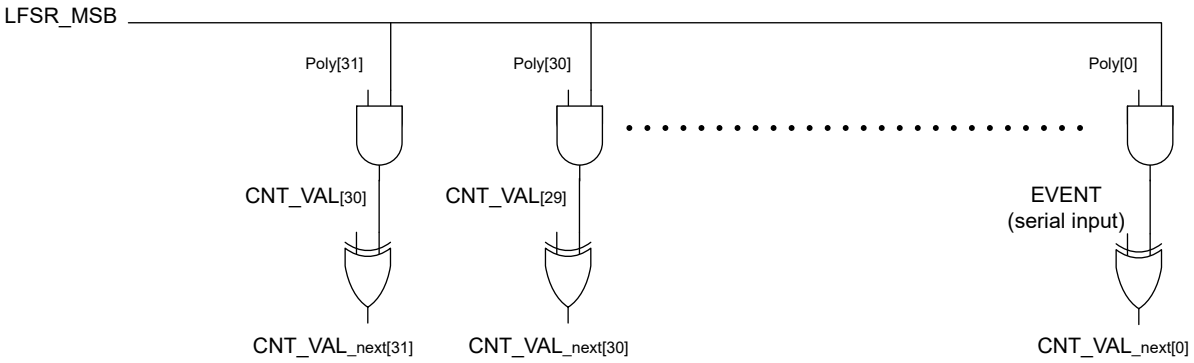


Figure 11-12. LFSR Modes

11.4.3 FSM Block

The Finite State Machine (FSM) block provides the ability to build programmable finite state machines with up to four states. The FSM block has two register bits and two external inputs, and can be programmed either as two 2-state machines or as a single 4-state machine. For additional flexibility, there are two auxiliary inputs (EXTRA_EXT_IN0 and EXTRA_EXT_IN1) that can be used to create larger combinational functions by giving up a state functionality. The structure of the FSM is shown in Figure 11-13.

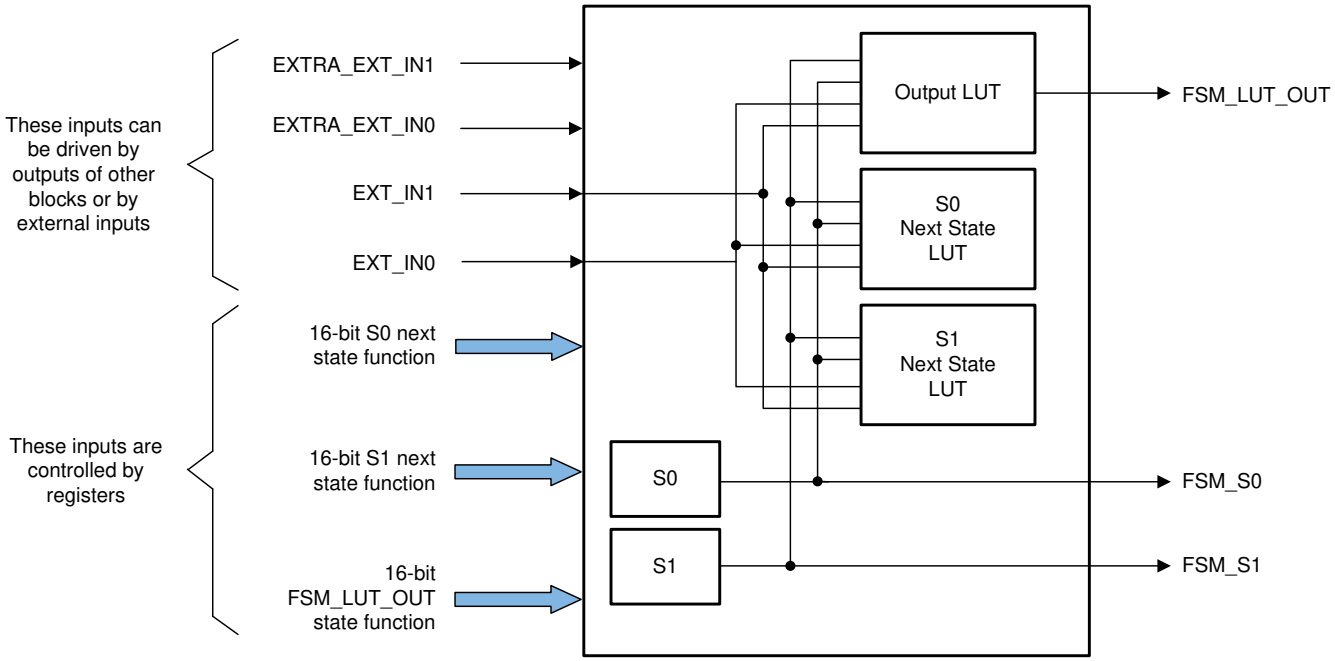


Figure 11-13. FSM Block

The signals and functionality of the FSM block are:

- **EXT_IN0 and EXT_IN1:** These are the two external inputs that can be used to control the output FSM_LUT_OUT or either of the states S0 and S1.
- **S0** and **S1** are two state bits that have independent state control equations.
- **16-bit S0 equation** defines a function (EXT_IN1, EXT_IN0, S1, S0). The four bits in the order defined are used as an index into the 16-bit register to decide the next state of S0.
- **16-bit S1 equation** defines a function (EXT_IN1, EXT_IN0, S1, S0). The four bits in the order defined are used as an index into the 16-bit register to decide the next state of S1.
- **16-bit output equation** defines a function (EXT_IN1, EXT_IN0, S1, S0). The four bits in the order defined are used as an index into the 16-bit register to decide the output value of FSM_LUT_OUT. An additional level of configurability is provided such that FSM_LUT_OUT can use extra inputs in case the states S0 and S1 are unused.

  One extra bit is used to select EXTRA_EXT_IN0 instead of S0. One extra bit is used to select EXTRA_EXT_IN1 instead of S1. Using these, one can effectively build 3-input or a 4-input LUT for the FSM_LUT_OUT by giving up one or two state bits, respectively.

The CFG_MISC_CTRL register controls the operation of the FSM block. Two bits in this register are used for each FSM Block to determine whether the FSM output LUT function uses the state variable S0/S1, or the corresponding extra external input signal FSM_EXTRA_EXT_INx. A 0 means use the state bit, and a 1 means use the FSM_EXTRA_EXT_IN0 / FSM_EXTRA_EXT_IN1 signal.

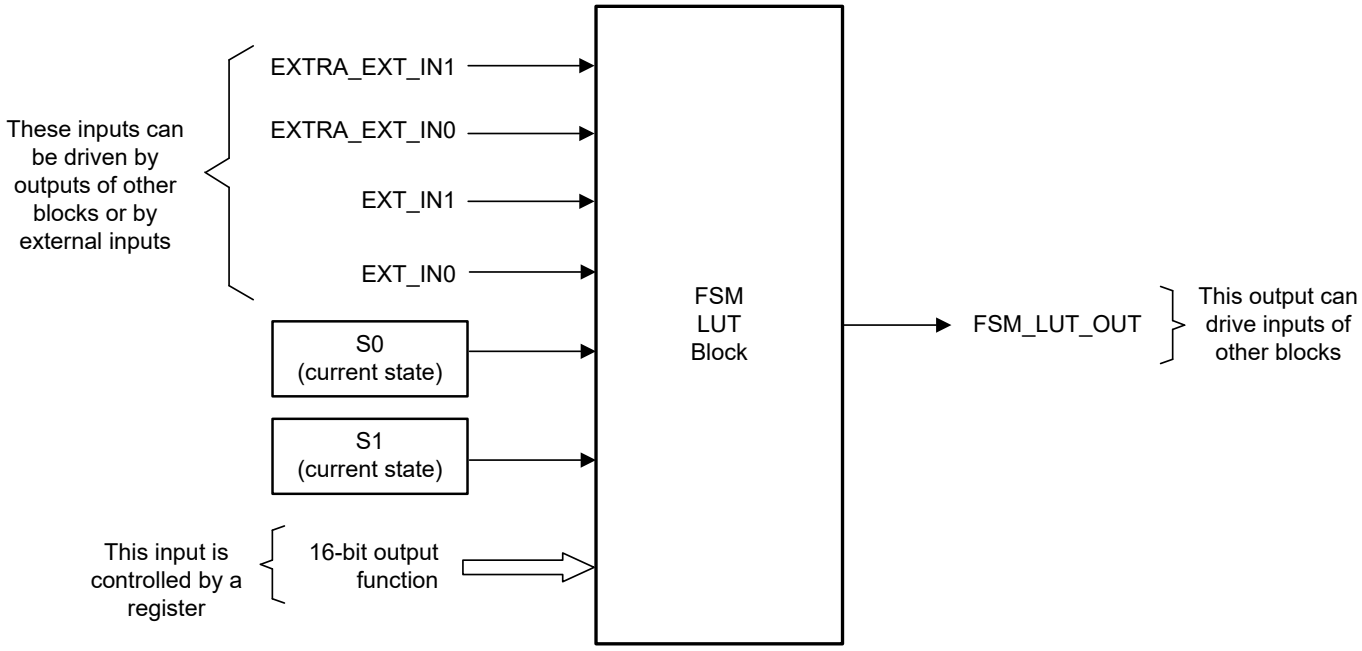


Figure 11-14. FSM LUT Block

11.4.4 LUT4 Block

This is a simple four input Look-Up table (LUT) block with inputs IN0, IN1, IN2, and IN3 (see Figure 11-15). Any combinatorial Boolean equation using the four inputs can be realized by programming the 16-bit control register associated with each LUT4 block. For more information, see the LUT4 register descriptions located in Section 11.10.

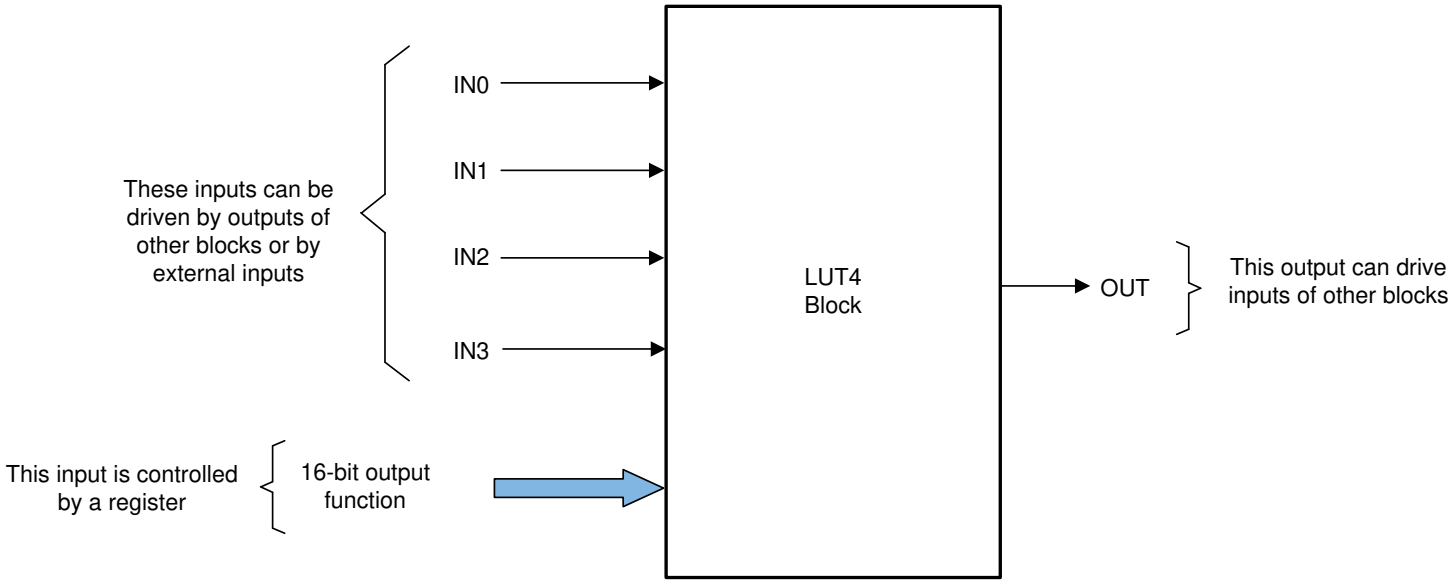


Figure 11-15. LUT4 Block

11.4.5 Output LUT Block

The output LUT block (Figure 11-16) is very similar in functionality to the LUT4 block, except that the output LUT block has three inputs. Unlike the other sub blocks, the outputs of these blocks are meant to go out of the tile and hence cannot be used by any other block within the tile. Any combinatorial function of the three inputs can be realized by the output LUT block. For more information, see the output LUT register descriptions located in Section 11.10.

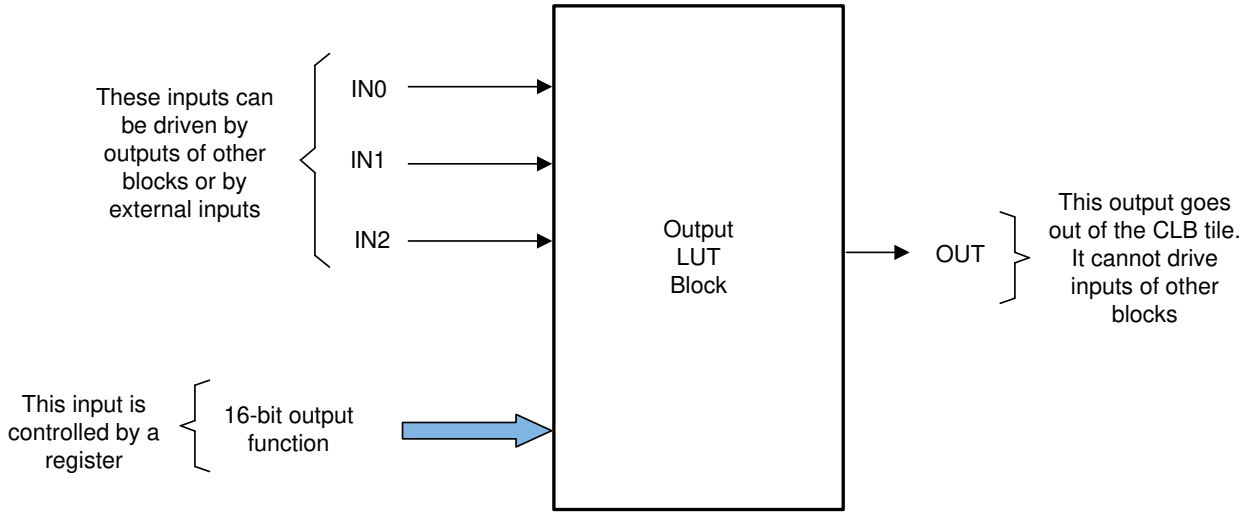


Figure 11-16. Output LUT Block

11.4.6 Asynchronous Output Conditioning (AOC) Block

The logic in the AOC block is organized into three stages with the inputs passing through different types of logic modification at each stage before proceeding to the next level. This is shown in Figure 11-17.

There are 8 inputs to this block. Each of these 8 inputs can pick the corresponding BOUNDARY input to the CLB or the CLB TILE output (for example, the INPUT 0 of the AOC block can choose between CLB BOUNDARY INPUT0 and CLB TILE OUTPUT 0). If the CLB TILE OUTPUT 0 is selected, the CLB TILE OUTPUT 0 is always registered before being sent to the subsequent asynchronous signal conditioning stages. In each of the three stages, there is always an option to do nothing and just send the signal as is to the next stage (bypass).

Stage 1: The input signal can be inverted before sending the signal to the next stage.

Stage 2: The signal coming from Stage 1 can be GATED with a gating control signal. The gating control signal can either be from a software register or can be any of the CLB TILE outputs. The GATING function can be a logical AND, OR, or XOR.

Stage 3: The input signal can be used to either set or clear the output on the rising edge of the signal. This is a purely asynchronous set/clear that occurs without needing any clocks. The release control signal, when high, restores the output to the default state (HIGH if asynchronous clear is selected and LOW if asynchronous set is selected). The release control signal can be either from a software register or can be any of the CLB TILE outputs. Optionally, instead of any of the asynchronous set and clear operations, the input signal can just be delayed by a clock cycle.

The interaction of the CLB TILE and the AOC block is shown in Figure 11-18.

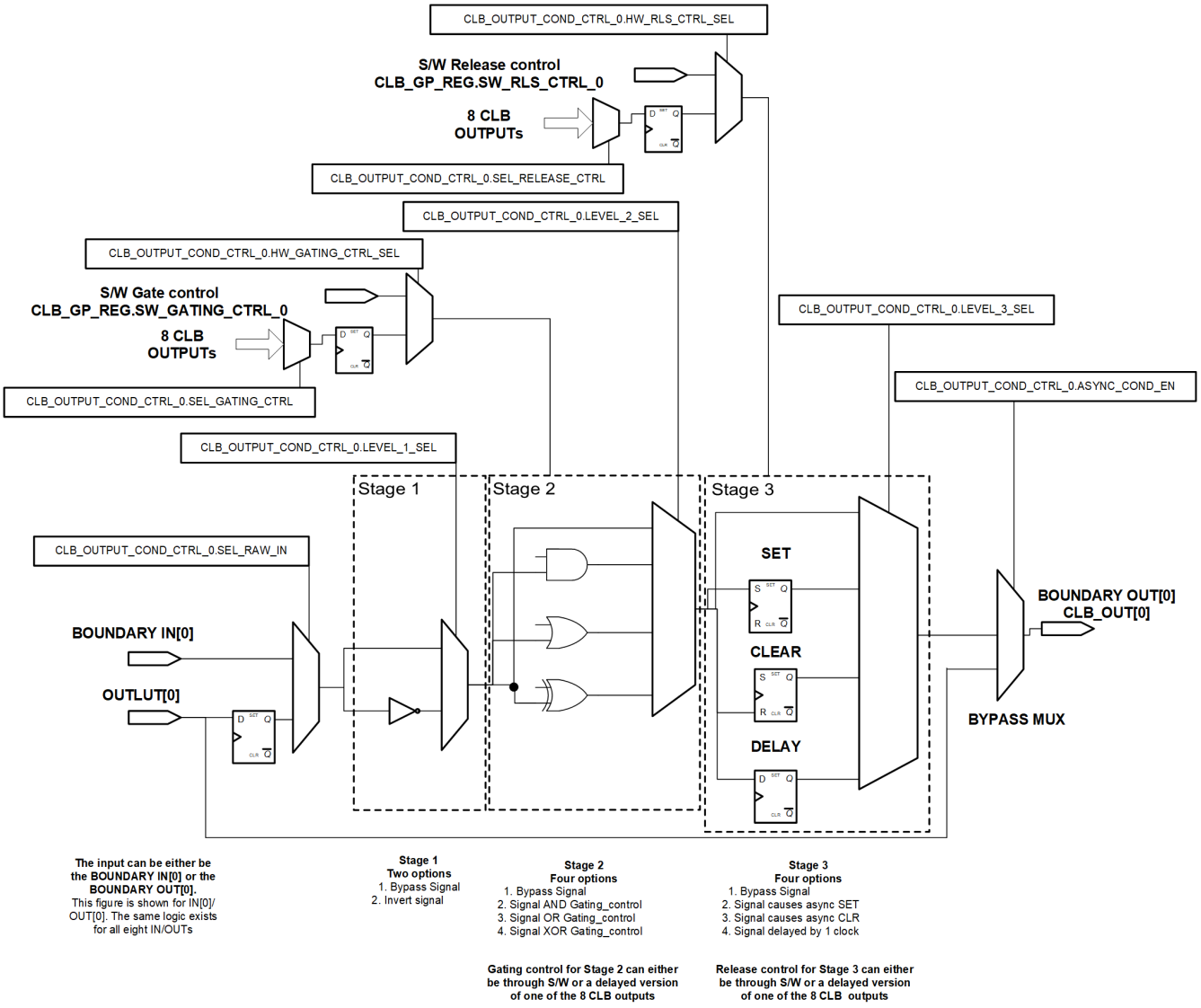


Figure 11-17. AOC Block

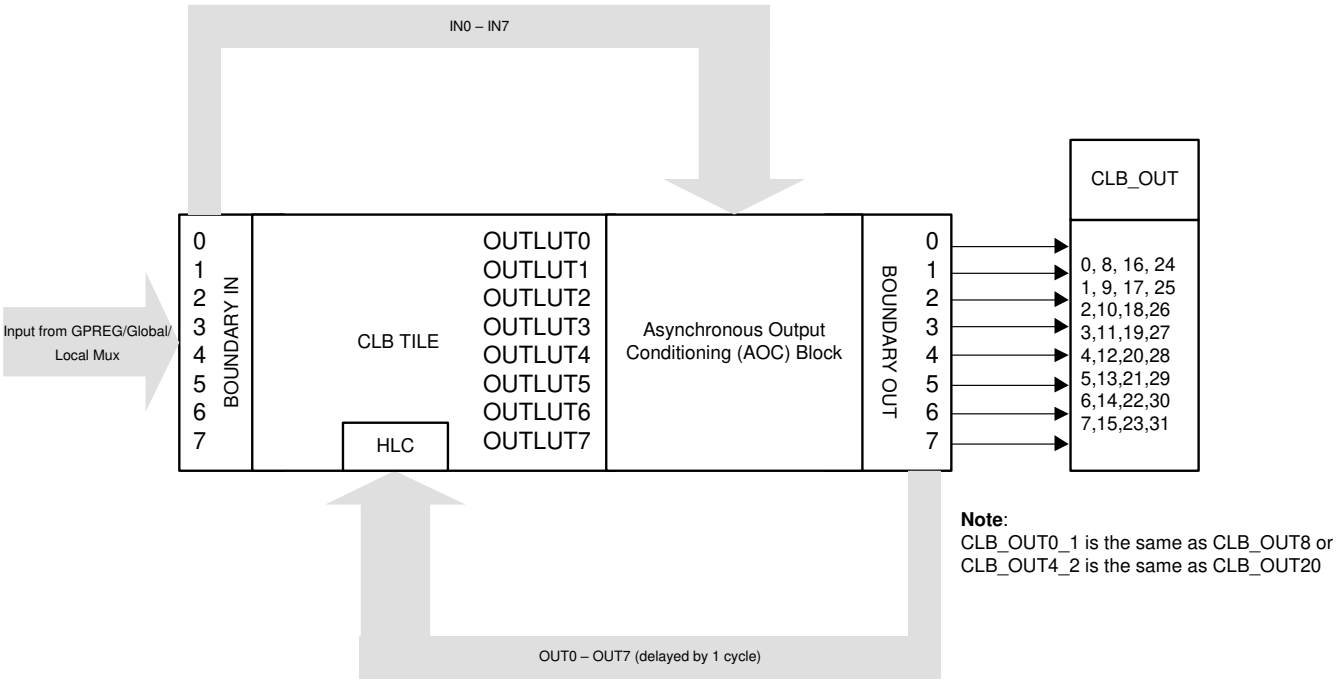


Figure 11-18. AOC Block and The CLB TILE

Note

Only CLB_OUT12 to CLB_OUT15 can be used as ASYNC outputs. This is the same as CLB_OUT4_1 to CLB_OUT7_1. GPIO Output XBAR can be used to route the OUT4_1 and OUT5_1 ASYNC outputs to GPIOs.

11.4.7 High Level Controller (HLC)

The High Level Controller (HLC) is significantly more complex than the other blocks in the CLB tile. The HLC performs two main functions:

- Provides a means of communication and data exchange with the rest of the device. This is done through two methods: a global access path to four general purpose HLC registers (R0 through R3) and PUSH and PULL FIFOs between the CPU and the HLC. The general-purpose HLC registers are designed to only be written to during device configuration time. To avoid unexpected behavior, the HLC registers must not be written to during run-time operation. The PUSH and PULL FIFOs are the primary avenues of data exchange during run-time, refer to Section 11.4.7.4 for more information.
- Provides a programmable, event-based action system, which performs computation, manipulation of logic functionality, and data movement. In other words, events can be configured to trigger a predefined set of actions in the CLB tile, or to initiate data exchange with the rest of the device.

The architecture of the HLC is shown in Figure 11-19. The HLC is an event-based system capable of handling up to four simultaneous events that can be selected from any outputs of the other blocks within the tile or from an external input.

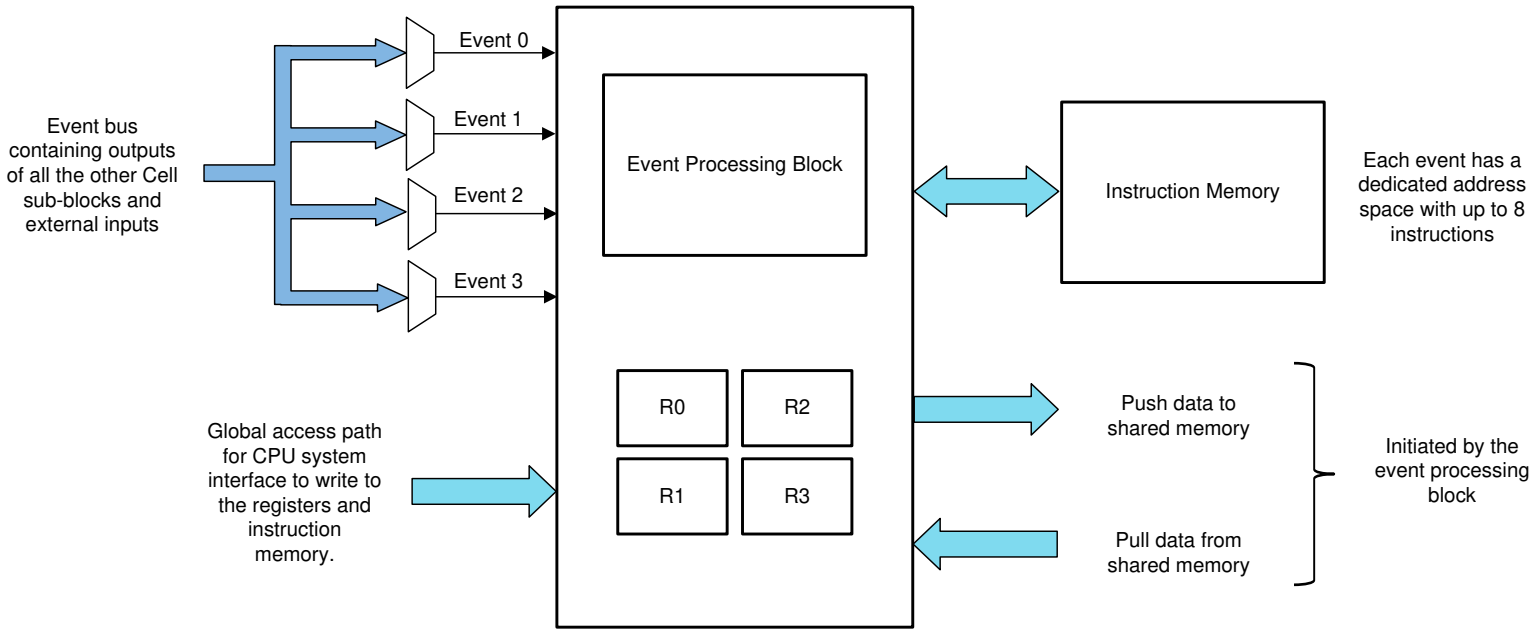


Figure 11-19. High Level Controller Block

11.4.7.1 High Level Controller Events

Each of the four HLC events has a dedicated address from which instructions are executed. Events are selected from the set of signals listed in Table 11-12. The lowest numbered event (Event 0) has the highest priority, and the highest numbered event (Event 3) has the lowest priority.

Table 11-12. HLC Event List

Index	HLC Event Mux
0	Always 0
1	COUNTER_0 MATCH2
2	COUNTER_0 ZERO
3	COUNTER_0 MATCH1
4	FSM_0 STATE_BIT_0
5	FSM_0 STATE_BIT_1
6	FSM_0 LUT output
7	LUT4_0 output
8	Always 1
9	COUNTER_1 MATCH2
10	COUNTER_1 ZERO
11	COUNTER_1 MATCH1
12	FSM_1 STATE_BIT_0
13	FSM_1 STATE_BIT_1
14	FSM_1 LUT output
15	LUT4_1 output
16	Always ‘0’
17	COUNTER_2 MATCH2
18	COUNTER_2 ZERO
19	COUNTER_2 MATCH1
20	FSM_2 STATE_BIT_0
21	FSM_2 STATE_BIT_1
22	FSM_2 LUT output
23	LUT4_2 output
24	External Input 0
25	External Input 1
26	External Input 2
27	External Input 3
28	External Input 4
29	External Input 5
30	External Input 6
31	External Input 7

Additional HLC inputs are available starting with Type 2 CLB and later. These inputs can be selected by choosing the alternate MUX options for the HLC module (HLC_ALT_MUX_SEL_n = 1) shown in Table 11-13.

Table 11-13. HLC ALT Event List

Index	HLC Event Mux
0	CLB_OUT_0
1	CLB_OUT_1
2	CLB_OUT_2
3	CLB_OUT_3
4	CLB_OUT_4
5	CLB_OUT_5
6	CLB_OUT_6
7	CLB_OUT_7
8	CLB_OUT_0.INVERTED
9	CLB_OUT_1.INVERTED
10	CLB_OUT_2.INVERTED
11	CLB_OUT_3.INVERTED
12	CLB_OUT_4.INVERTED
13	CLB_OUT_5.INVERTED
14	CLB_OUT_6.INVERTED
15	CLB_OUT_7.INVERTED
16	CLB_ASYNC_OUT_0
17	CLB_ASYNC_OUT_1
18	CLB_ASYNC_OUT_2
19	CLB_ASYNC_OUT_3
20	CLB_ASYNC_OUT_4
21	CLB_ASYNC_OUT_5
22	CLB_ASYNC_OUT_6
23	CLB_ASYNC_OUT_7
24	CLB_ASYNC_OUT_0.INVERTED
25	CLB_ASYNC_OUT_1.INVERTED
26	CLB_ASYNC_OUT_2.INVERTED
27	CLB_ASYNC_OUT_3.INVERTED
28	CLB_ASYNC_OUT_4.INVERTED
29	CLB_ASYNC_OUT_5.INVERTED
30	CLB_ASYNC_OUT_6.INVERTED
31	CLB_ASYNC_OUT_7.INVERTED

11.4.7.2 High Level Controller Instructions

The instruction memory supports up to eight instructions per event. Each instruction sequence gets triggered on the rising edge of the corresponding event. Starting with CLB Type 2, the option to trigger the execution of instructions using both falling edge and rising edge is available.

The HLC memory supports up to eight instructions per event, starting at the beginning of the fixed address range shown in Table 11-14. An instruction sequence is triggered on the rising edge of the corresponding event. If two or more events occur simultaneously, the associated instruction sequences each are executed sequentially in priority order.

Table 11-14. HLC Instruction Address Ranges

Address	Instructions for
00000 to 00111	Event 0
01000 to 01111	Event 1
10000 to 10111	Event 2
11000 to 11111	Event 3

The HLC instruction format is shown in Table 11-15.

Table 11-15. HLC Instruction Format

Last Instruction Bit	5-Bit Opcode	3-Bit Source	3-Bit Destination
This bit when set to 1 stops execution after the current instruction.	MOV 00000 MOV_T1 00001 MOV_T2 00010 PUSH 00011 PULL 00100 ADD 00101 SUB 00110 INTR 00111	Source can be R0, R1, R2, R3, C0, C1, C2.	Destination can be R0, R1, R2, R3, C0, C1, C2. Note that for ADD/SUB instructions, only R0, R1, R2, or R3 can be the destination.

R0, R1, R2, and R3 are four 32-bit general-purpose registers in the HLC. C0, C1, and C2 are three counter registers present in the CLB tile. <Src> is used to indicate the source and <Dest> is used to indicate the destination. Table 11-16 describes the HLC instructions.

Table 11-16. HLC Instruction Description

Instruction	Description
ADD <Src>, <Dest>	This instruction performs an unsigned 32-bit addition. <Dest> = <Dest> + <Src>. The <Src> can be R0, R1, R2, R3, C0, C1, or C2. The <Dest> can only be R0, R1, R2, or R3.
INTR <6-bit constant>	This instruction flags an interrupt through the CPU interface. The 6-bit constant is stored in the interrupt flag register CLB_INTR_TAG_REG. If multiple INTR instructions are called consecutively, only the first one has an effect. When multiple INTR calls are needed, each can be separated by other HLC instructions to make sure the interrupt calls take effect. **Note** Starting with CLB Type 2, NMI can be generated by the CLB. This feature is DISABLED by default and must be enabled (**CLB_LOAD_EN.NMI_EN**).
MOV <Src>, <Dest>	This instruction moves <Src> to <Dest>. Both <Src> and <Dest> can be any of R0, R1, R2, R3, C0, C1, or C2. The COUNT_EVENT_CTRL_x bit must be configured to load (that is, 0) for indirect loads and HLC loads of the counter to take effect.

Table 11-16. HLC Instruction Description (continued)

Instruction	Description
MOV_T1 <Src>, <Dest>	This instruction moves <Src> to the Match1 register of the <Dest> counter. <Src> can be any of the registers R0, R1, R2, R3, or the counter values associated with C0, C1, or C2. <Dest> is the Match1 register of any of the counters C0, C1, or C2. Examples are: • This instruction moves the count value in C1 into register R0: MOV_T1 C1 R0 • This instruction moves the value in R2 into the Match1 register of counter C0: MOV_T1 R2 C0
MOV_T2 <Src>, <Dest>	This instruction is similar to MOV_T1. The instruction moves <Src> to the Match2 register of the <Dest> counter. <Src> can be any of the registers R0, R1, R2, R3, or the counter values associated with C0, C1, or C2. <Dest> is the Match2 register of any of the counters C0, C1, or C2.
PULL <Dest>	This instruction transfers data from the data exchange pull memory buffer in the CPU interface to the <Dest> register. <Dest> can be any of R0, R1, R2, or R3. The PULL instruction is used as seen from the High Level Controller and a PULL operation reads (pulls) data from an internal 4-word FIFO.
PUSH <Src>	This instruction transfers data from <Src> to the data exchange push memory buffer in the CPU interface. <Src> can be any of R0, R1, R2, R3, C0, C1, or C2. The PUSH instruction is used as seen from the High Level Controller and pushes data into an internal 4 word FIFO.
SUB <Src>, <Dest>	This instruction performs an unsigned 32-bit subtraction. <Dest> = <Dest> - <Src>. The <Src> can be R0, R1, R2, R3, C0, C1, or C2. The <Dest> can only be R0, R1, R2, or R3.

MOV, MOV_T1, MOV_T2, ADD, SUB, and INTR instructions take one cycle to execute. PUSH and PULL require two cycles to execute. Note that the PUSH and PULL instructions are pipeline protected, meaning that a register can be used immediately after a PUSH/PULL to that register.

For multiple events triggered simultaneously, if the last instruction in the higher priority event is a PUSH or a PULL, there is an additional cycle delay between the end of the higher priority event and the start of the next event. If the last instruction is not a PUSH or PULL, then there is no cycle delay between the events.

11.4.7.3 <Src> and <Dest>

Three bits are used to encode the <Src> and <Dest> registers as shown in Table 11-17.

Table 11-17. HLC Register Encoding

Bits	Register
000	R0
001	R1
010	R2
011	R3
100	C0
101	C1
110	C2

11.4.7.4 Operation of the PUSH and PULL Instructions (Overflow and Underflow Detection)

The PUSH and PULL operations of the HLC are intended for data exchange with the host system. There are separate FIFO buffers for PUSH and PULL operations. For example, a series of PUSH operations write to successive locations in a linearly mapped-memory buffer. The PUSH buffer and the PULL buffer are mapped at address offsets shown in the *Registers* section.

The CPU can read from and write to the PUSH and PULL buffers, respectively, to exchange data with the HLC. Data pushed by the HLC is read by the CPU from the PUSH buffers. Data sent from the CPU to the HLC is written by the CPU to the PULL buffer and is read by the HLC using the PULL instruction.

Refer to *clb_ex13_push_pull* for guidance on properly using the PUSH and PULL buffers. To make use of one of the CLB inputs as a GPREG, have this input indicate when data is written to the FIFO by the CPU.

There are separate PUSH and PULL address pointers that increment each time the HLC performs a PUSH or PULL operation. These address pointers are also memory-mapped so that the CPU can determine the value. These address pointers are also writable and can be reset by the CPU at any time.

Overflow and underflow detection is done by simply reading the values of the PUSH and PULL address pointers.

In the CLB module of the device, the depth of the PUSH and PULL FIFOs is four 32-bit words each. If the CPU starts a fresh data transfer to the PULL buffers and sees the address pointer greater than four, then an underflow has occurred since the HLC has pulled more data than the number of words written by the CPU into the buffer.

11.5 CPU Interface

11.5.1 Register Description

There are three classes of registers that are used to control and configure the CLB tile. This specification only describes the offset addresses of the registers. The absolute register addresses are different for each CLB tile. The three instances of the various blocks (LUT4, FSM, and Counter Block) are numbered 0, 1, and 2.

- **Logic configuration registers (0x000–0x0FF):** These registers control the core reconfiguration logic for the tile. All registers in this group are EALLOW protected and also protected by the LOCK register.
- **Top level control registers (0x100–0x1FF):** These registers are used for top level and device related control of the CLB. These registers typically control mux selects for inputs, global enables, and so forth, and are accessible by normal memory mapped access. Some of these registers have EALLOW and LOCK protection.
- **Data exchange registers (0x200–0x3FF):** These registers are used to exchange data between the CLB and the rest of the device. The registers are accessible by normal memory-mapped access and no EALLOW or LOCK protection exists.

Note

EALLOW protection means that the write access to the register is enabled only when the EALLOW instruction has been executed prior to the write access. The complementary EDIS instruction disables access to all registers protected in this way. For more information, see Section 11.10.

11.5.2 Non-Memory Mapped Registers

The memory-mapped CLB registers are described later in this chapter; however, many of the CLB resources including counters, the instruction memory of the High Level Controller, and the HLC general-purpose registers (R0 through R3) are only indirectly accessible through a local interface bus and are not memory-mapped. These registers are accessible through the two memory-mapped registers CLB_LOAD_DATA and CLB_LOAD_ADDR. Note that the general-purpose registers R0 through R3 must only be written to during configuration-time and are not intended to be written to during run-time. Writes during run-time can lead to unexpected behavior. If run-time data exchange is desired, refer to Section 11.4.7.4.

Load the data to be written into the CLB_LOAD_DATA register, then load the appropriate address into CLB_LOAD_ADDR to determine where this data is written. Writing a 1 to bit position 0 in the CLB_LOAD_EN register then causes an internal write operation to be triggered. The address allocation for the CLB_LOAD_ADDR register is shown in Table 11-18.

Note

The COUNT_EVENT_CTRL_x bit must be configured to load (that is, 0) for indirect loads and for HLC loads of the counter to take effect.

Table 11-18. Non-Memory Mapped Register Addresses

Address (Binary)	Resource
000000 to 000010	Counter 0 to 2 Load value
000100 to 000110	Counter 0 to 2 Match1 value
001000 to 001010	Counter 0 to 2 Match2 value
001100 to 001111	R0 to R3 of High Level Controller
100000 to 100111	Instructions for Event 0
101000 to 101111	Instructions for Event 1
110000 to 110111	Instructions for Event 2
111000 to 111111	Instructions for Event 3

Use the following steps to load the value 0x11223344 into the general purpose R0 register:

1. Write 0x11223344 to CLB_LOAD_DATA.
2. Write 0xc to CLB_LOAD_ADDR.
3. Write 0x1 to CLB_LOAD_EN.

Note

Even though HLC registers are accessible by the CPU, your application code needs to make sure that no other CLB internal logics are updating the same HLC register at the same time, causing a race condition.

11.6 RTDMA Access

The RTDMA does not have access to the CLB memory-mapped registers, including the PUSH and PULL FIFO registers. For more information, refer to Section 11.10.

11.7 CLB Data Export Through SPI RX Buffer

For a continuous export of data from the CLB peripherals, SPI RX buffers can be used. CLB data can be exported through the SPI RX buffers without CPU/CLA interventions.

CLB1 to CLB5 have access to SPIA to SPIE, respectively, as shown in Table 11-19.

Table 11-19. CLB to SPI RX Access

CLB Instance	SPI Instance
CLB1	SPIA
CLB2	SPIB
CLB3	SPIC
CLB4	SPID
CLB5	SPIE

When the CLB to SPI data exporting is enabled, 16-bit data can be exported from CLB to SPI RX buffers. The 32-bit HLC R0 register is the data that is exported to the SPI RX buffers. The user can select which 16-bit range of the HLC R0 is exported by configuring the CLB_SPI_DATA_CTRL_HI.SHIFT. The CLB also controls when HLC R0 data must be transferred to the SPI RX buffer through CLB_SPI_DATA_CTRL_HI.STRB that selects one of the HLC event signals from the static switch block.

When CLB to SPI data exporting is required, note:

- The selected SPI transmit functionality is not affected.
- Even though the data is being pushed into the SPI RX buffers by the CLB, the SPI RX interrupt and the RTDMA trigger for SPI RX in the respective peripherals must be configured.
- The SPI can resume normal operation when the CLB to SPI data exporting is disabled.

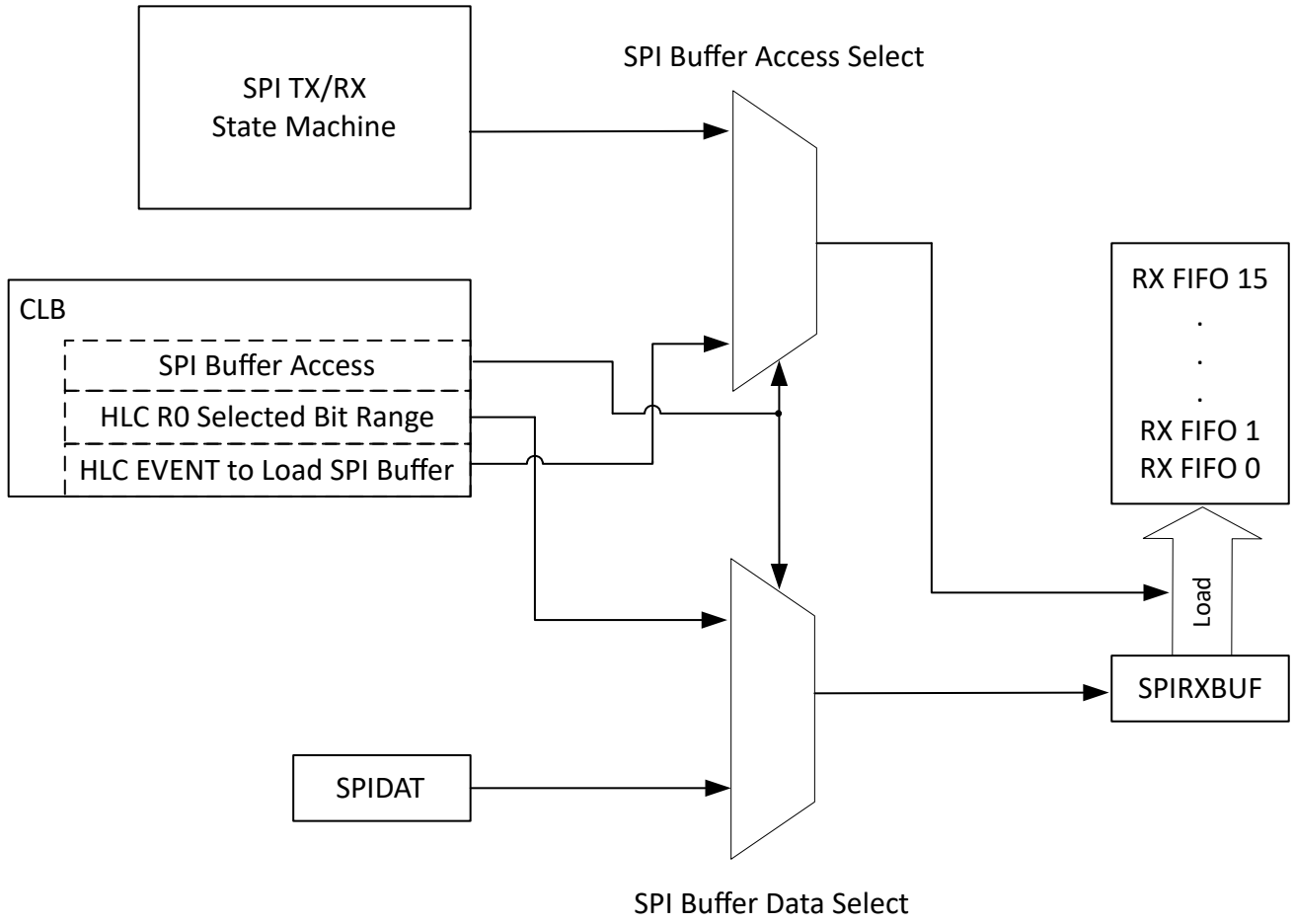


Figure 11-20. CLB Control of SPI RX Buffer

11.8 CLB Pipeline Mode

When operating the CLB TILE at frequencies higher than 100MHz, the Pipeline mode MUST be enabled. When CLB Pipeline mode is enabled, the operations of the HLC and COUNTER modules are changed.

- When operating at higher frequencies that require Pipeline mode to be enabled, the pipelined versions of the the CLB CELL OUTPUTs are brought into the HLC. The Pipeline mode causes the CELL outputs delayed by 1 clock cycle to be used as the source of the HLC event triggers.
- In Pipeline mode, the COUNTER module's add/sub/shift operations, which are triggered by an event, use the value of the counter in the previous clock cycle (pipelined).

To enable the CLB Pipeline mode, set the CLB_LOAD_EN.PIPELINE_EN.

11.9 Software

11.9.1 CLB Registers to Driverlib Functions

Table 11-20. CLB Registers to Driverlib Functions

File	Driverlib Function
COUNT_RESET	
clb.h	CLB_selectCounterInputs
COUNT_MODE_1	
clb.h	CLB_selectCounterInputs
COUNT_MODE_0	
clb.h	CLB_selectCounterInputs
COUNT_EVENT	
clb.h	CLB_selectCounterInputs
FSM_EXTRA_IN0	
clb.h	CLB_selectFSMInputs
FSM_EXTERNAL_IN0	
clb.h	CLB_selectFSMInputs
FSM_EXTERNAL_IN1	
clb.h	CLB_selectFSMInputs
FSM_EXTRA_IN1	
clb.h	CLB_selectFSMInputs
LUT4_IN0	
clb.h	CLB_selectLUT4Inputs
LUT4_IN1	
clb.h	CLB_selectLUT4Inputs
LUT4_IN2	
clb.h	CLB_selectLUT4Inputs
LUT4_IN3	
clb.h	CLB_selectLUT4Inputs
FSM_LUT_FN1_0	
clb.h	CLB_configFSMLUTFunction
FSM_LUT_FN2	
clb.h	CLB_configFSMLUTFunction
LUT4_FN1_0	
clb.h	CLB_configLUT4Function
LUT4_FN2	
clb.h	CLB_configLUT4Function
FSM_NEXT_STATE_0	
clb.h	CLB_configFSMNextState
FSM_NEXT_STATE_1	
clb.h	CLB_configFSMNextState
FSM_NEXT_STATE_2	
clb.h	CLB_configFSMNextState
MISC_CONTROL	
clb.h	CLB_configMiscCtrlModes
OUTPUT_LUT_0	
clb.h	CLB_configOutputLUT

Table 11-20. CLB Registers to Driverlib Functions (continued)

File	Driverlib Function
OUTPUT_LUT_1	
-	See OUTPUT_LUT_0
OUTPUT_LUT_2	
-	See OUTPUT_LUT_0
OUTPUT_LUT_3	
-	See OUTPUT_LUT_0
OUTPUT_LUT_4	
-	See OUTPUT_LUT_0
OUTPUT_LUT_5	
-	See OUTPUT_LUT_0
OUTPUT_LUT_6	
-	See OUTPUT_LUT_0
OUTPUT_LUT_7	
-	See OUTPUT_LUT_0
HLC_EVENT_SEL	
clb.h	CLB_configHLCEventSelect
COUNT_MATCH_TAP_SEL	
clb.h	CLB_configCounterTapSelects
OUTPUT_COND_CTRL_0	
clb.h	CLB_configAOC
OUTPUT_COND_CTRL_1	
-	
OUTPUT_COND_CTRL_2	
-	
OUTPUT_COND_CTRL_3	
-	
OUTPUT_COND_CTRL_4	
-	
OUTPUT_COND_CTRL_5	
-	
OUTPUT_COND_CTRL_6	
-	
OUTPUT_COND_CTRL_7	
-	
MISC_ACCESS_CTRL	
clb.h	CLB_disableOutputMaskUpdates
clb.h	CLB_enableOutputMaskUpdates
clb.h	CLB_disableSPIBufferAccess
clb.h	CLB_enableSPIBufferAccess
SPI_DATA_CTRL_HI	
clb.h	CLB_configSPIBufferLoadSignal
clb.h	CLB_configSPIBufferShift
clb.h	CLB_enableSPIStrobeDelay
clb.h	CLB_disableSPIStrobeDelay
LOAD_EN	

Table 11-20. CLB Registers to Driverlib Functions (continued)

File	Driverlib Function
clb.h	CLB_enableCLB
clb.h	CLB_disableCLB
clb.h	CLB_enableNMI
clb.h	CLB_disableNMI
clb.h	CLB_writeInterface
clb.h	CLB_enablePipelineMode
clb.h	CLB_disablePipelineMode
LOAD_ADDR	
clb.h	CLB_writeInterface
LOAD_DATA	
clb.h	CLB_writeInterface
INPUT_FILTER	
clb.h	CLB_selectInputFilter
clb.h	CLB_enableSynchronization
clb.h	CLB_disableSynchronization
clb.h	CLB_enableInputPipelineMode
clb.h	CLB_disableInputPipelineMode
IN_MUX_SEL_0	
clb.h	CLB_configGPInputMux
LCL_MUX_SEL_1	
clb.h	CLB_configLocalInputMux
LCL_MUX_SEL_2	
clb.h	CLB_configLocalInputMux
BUF_PTR	
clb.c	CLB_clearFIFOs
GP_REG	
clb.h	CLB_writeSWReleaseControl
clb.h	CLB_writeSWGateControl
clb.h	CLB_setGPREG
clb.h	CLB_getGPREG
OUT_EN	
clb.h	CLB_setOutputMask
GLBL_MUX_SEL_1	
clb.h	CLB_configGlobalInputMux
GLBL_MUX_SEL_2	
clb.h	CLB_configGlobalInputMux
PRESCALE_CTRL	
clb.h	CLB_configureClockPrescalar
clb.h	CLB_configureStrobeMode
INTR_TAG_REG	
clb.h	CLB_getInterruptTag
clb.h	CLB_clearInterruptTag
LOCK	
clb.h	CLB_enableLock
HLC_INSTR_READ_PTR	

Table 11-20. CLB Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
HLC_INSTR_VALUE	
-	
DBG_OUT_2	
-	
DBG_R0	
-	
DBG_R1	
-	
DBG_R2	
-	
DBG_R3	
-	
DBG_C0	
-	
DBG_C1	
-	
DBG_C2	
-	
DBG_OUT	
clb.h	CLB_getOutputStatus
PUSH(i)	
clb.c	CLB_readFIFOs
PULL(i)	
clb.c	CLB_clearFIFOs
clb.c	CLB_writeFIFOs

11.9.2 CLB Examples

NOTE: These examples are located in the C2000Ware installation at the following location: C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/clb

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

11.10 CLB Registers

This Section describes the CLB Registers.

11.10.1 CLB Base Address Table

Table 11-21. CLB Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
CLB_LOGIC_CONFIG_REGS	CLB1_LOGICCFG_BASE	0x7012_0000	YES	YES	YES	YES	YES	YES	-	YES
CLB_LOGIC_CONTROL_REGS	CLB1_LOGICCTRL_BASE	0x7012_0200	YES	YES	YES	YES	YES	YES	-	YES
CLB_DATA_EXCHANGE_REGS	CLB1_DATAEXCH_BASE	0x7012_0300	YES	YES	YES	YES	YES	YES	-	YES
CLB_LOGIC_CONFIG_REGS	CLB2_LOGICCFG_BASE	0x7012_1000	YES	YES	YES	YES	YES	YES	-	YES

Table 11-21. CLB Base Address Table (continued)

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
CLB_LOGIC_CONTROL_REGS	CLB2_LOGICCTRL_BASE	0x7012_1200	YES	YES	YES	YES	YES	YES	-	YES
CLB_DATA_EXCHANGE_REGS	CLB2_DATAEXCH_BASE	0x7012_1300	YES	YES	YES	YES	YES	YES	-	YES
CLB_LOGIC_CONFIG_REGS	CLB3_LOGICCFG_BASE	0x7012_2000	YES	YES	YES	YES	YES	YES	-	YES
CLB_LOGIC_CONTROL_REGS	CLB3_LOGICCTRL_BASE	0x7012_2200	YES	YES	YES	YES	YES	YES	-	YES
CLB_DATA_EXCHANGE_REGS	CLB3_DATAEXCH_BASE	0x7012_2300	YES	YES	YES	YES	YES	YES	-	YES
CLB_LOGIC_CONFIG_REGS	CLB4_LOGICCFG_BASE	0x7012_3000	YES	YES	YES	YES	YES	YES	-	YES
CLB_LOGIC_CONTROL_REGS	CLB4_LOGICCTRL_BASE	0x7012_3200	YES	YES	YES	YES	YES	YES	-	YES
CLB_DATA_EXCHANGE_REGS	CLB4_DATAEXCH_BASE	0x7012_3300	YES	YES	YES	YES	YES	YES	-	YES
CLB_LOGIC_CONFIG_REGS	CLB5_LOGICCFG_BASE	0x7012_4000	YES	YES	YES	YES	YES	YES	-	YES
CLB_LOGIC_CONTROL_REGS	CLB5_LOGICCTRL_BASE	0x7012_4200	YES	YES	YES	YES	YES	YES	-	YES
CLB_DATA_EXCHANGE_REGS	CLB5_DATAEXCH_BASE	0x7012_4300	YES	YES	YES	YES	YES	YES	-	YES
CLB_LOGIC_CONFIG_REGS	CLB6_LOGICCFG_BASE	0x7012_5000	YES	YES	YES	YES	YES	YES	-	YES
CLB_LOGIC_CONTROL_REGS	CLB6_LOGICCTRL_BASE	0x7012_5200	YES	YES	YES	YES	YES	YES	-	YES
CLB_DATA_EXCHANGE_REGS	CLB6_DATAEXCH_BASE	0x7012_5300	YES	YES	YES	YES	YES	YES	-	YES

11.10.2 CLB_LOGIC_CONFIG_REGS Registers

Table 11-22 lists the memory-mapped registers for the CLB_LOGIC_CONFIG_REGS registers. All register offset addresses not listed in Table 11-22 should be considered as reserved locations and the register contents should not be modified.

Table 11-22. CLB_LOGIC_CONFIG_REGS Registers

Offset	Acronym	Register Name	Protection
4h	CLB_COUNT_RESET	Counter Block RESET	LOCK
8h	CLB_COUNT_MODE_1	Counter Block MODE_1	LOCK
Ch	CLB_COUNT_MODE_0	Counter Block MODE_0	LOCK
10h	CLB_COUNT_EVENT	Counter Block EVENT	LOCK
14h	CLB_FSM_EXTRA_IN0	FSM Extra EXT_IN0	LOCK
18h	CLB_FSM_EXTERNAL_IN0	FSM EXT_IN0	LOCK
1Ch	CLB_FSM_EXTERNAL_IN1	FSM_EXT_IN1	LOCK
20h	CLB_FSM_EXTRA_IN1	FSM Extra_EXT_IN1	LOCK
24h	CLB_LUT4_IN0	LUT4_0/1/2 IN0 input source	LOCK
28h	CLB_LUT4_IN1	LUT4_0/1/2 IN1 input source	LOCK
2Ch	CLB_LUT4_IN2	LUT4_0/1/2 IN2 input source	LOCK
30h	CLB_LUT4_IN3	LUT4_0/1/2 IN3 input source	LOCK
38h	CLB_FSM_LUT_FN1_0	LUT function for FSM Unit 1 and Unit 0	LOCK
3Ch	CLB_FSM_LUT_FN2	LUT function for FSM Unit 2	LOCK
40h	CLB_LUT4_FN1_0	LUT function for LUT4 block of Unit 1 and 0	LOCK
44h	CLB_LUT4_FN2	LUT function for LUT4 block of Unit 2	LOCK
48h	CLB_FSM_NEXT_STATE_0	FSM Next state equations for Unit 0	LOCK
4Ch	CLB_FSM_NEXT_STATE_1	FSM Next state equations for Unit 1	LOCK
50h	CLB_FSM_NEXT_STATE_2	FSM Next state equations for Unit 2	LOCK
54h	CLB_MISC_CONTROL	Static controls for Ctr,FSM	LOCK
58h	CLB_OUTPUT_LUT_0	Inp Sel, LUT fns for Out0	LOCK
5Ch	CLB_OUTPUT_LUT_1	Inp Sel, LUT fns for Out1	LOCK
60h	CLB_OUTPUT_LUT_2	Inp Sel, LUT fns for Out2	LOCK
64h	CLB_OUTPUT_LUT_3	Inp Sel, LUT fns for Out3	LOCK
68h	CLB_OUTPUT_LUT_4	Inp Sel, LUT fns for Out4	LOCK
6Ch	CLB_OUTPUT_LUT_5	Inp Sel, LUT fns for Out5	LOCK
70h	CLB_OUTPUT_LUT_6	Inp Sel, LUT fns for Out6	LOCK
74h	CLB_OUTPUT_LUT_7	Inp Sel, LUT fns for Out7	LOCK
78h	CLB_HLC_EVENT_SEL	Event Selector register for the High Level controller	LOCK
7Ch	CLB_COUNT_MATCH_TAP_SEL	Counter tap values for match1 and match2 outputs	LOCK
80h	CLB_OUTPUT_COND_CTRL_0	Output conditioning control for output 0	LOCK
84h	CLB_OUTPUT_COND_CTRL_1	Output conditioning control for output 1	LOCK
88h	CLB_OUTPUT_COND_CTRL_2	Output conditioning control for output 2	LOCK
8Ch	CLB_OUTPUT_COND_CTRL_3	Output conditioning control for output 3	LOCK
90h	CLB_OUTPUT_COND_CTRL_4	Output conditioning control for output 4	LOCK
94h	CLB_OUTPUT_COND_CTRL_5	Output conditioning control for output 5	LOCK
98h	CLB_OUTPUT_COND_CTRL_6	Output conditioning control for output 6	LOCK
9Ch	CLB_OUTPUT_COND_CTRL_7	Output conditioning control for output 7	LOCK
A0h	CLB_MISC_ACCESS_CTRL	Miscellaneous Access and enable control	LOCK
A2h	CLB_SPI_DATA_CTRL_HI	CLB to SPI buffer control High	LOCK

Complex bit access types are encoded to fit into small table cells. Table 11-23 shows the codes that are used for access types in this section.

Table 11-23. CLB_LOGIC_CONFIG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

11.10.2.1 CLB_COUNT_RESET Register (Offset = 4h) [Reset = 00000000h]

CLB_COUNT_RESET is shown in Figure 11-21 and described in Table 11-24.

Return to the Summary Table.

Counter Block RESET

Figure 11-21. CLB_COUNT_RESET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	SEL_2					SEL_1					SEL_0				
R/W1C-0h																	R/W-0h					R/W-0h					R/W-0h				

Table 11-24. CLB_COUNT_RESET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14-10	SEL_2	R/W	0h	Counter reset select inputs for unit 2. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	SEL_1	R/W	0h	Counter reset select inputs for unit 1. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	SEL_0	R/W	0h	Counter reset select inputs for unit 0. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.2 CLB_COUNT_MODE_1 Register (Offset = 8h) [Reset = 00000000h]

CLB_COUNT_MODE_1 is shown in Figure 11-22 and described in Table 11-25.

Return to the Summary Table.

Counter Block MODE_1

Figure 11-22. CLB_COUNT_MODE_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	SEL_2					SEL_1					SEL_0				
R/W1C-0h																	R/W-0h					R/W-0h					R/W-0h				

Table 11-25. CLB_COUNT_MODE_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14-10	SEL_2	R/W	0h	Counter MODE_1 select inputs for unit 2. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	SEL_1	R/W	0h	Counter MODE_1 select inputs for unit 1. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	SEL_0	R/W	0h	Counter MODE_1 select inputs for unit 0. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.3 CLB_COUNT_MODE_0 Register (Offset = Ch) [Reset = 00000000h]

CLB_COUNT_MODE_0 is shown in Figure 11-23 and described in Table 11-26.

Return to the Summary Table.

Counter Block MODE_0

Figure 11-23. CLB_COUNT_MODE_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	SEL_2					SEL_1					SEL_0				
R/W1C-0h																	R/W-0h					R/W-0h					R/W-0h				

Table 11-26. CLB_COUNT_MODE_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14-10	SEL_2	R/W	0h	Counter MODE_0 select inputs for unit 2. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	SEL_1	R/W	0h	Counter MODE_0 select inputs for unit 1. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	SEL_0	R/W	0h	Counter MODE_0 select inputs for unit 0. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.4 CLB_COUNT_EVENT Register (Offset = 10h) [Reset = 00000000h]

CLB_COUNT_EVENT is shown in Figure 11-24 and described in Table 11-27.

Return to the Summary Table.

Counter Block EVENT

Figure 11-24. CLB_COUNT_EVENT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	SEL_2					SEL_1					SEL_0				
R/W1C-0h																	R/W-0h					R/W-0h					R/W-0h				

Table 11-27. CLB_COUNT_EVENT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14-10	SEL_2	R/W	0h	Counter event select inputs for unit 2. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	SEL_1	R/W	0h	Counter event select inputs for unit 1. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	SEL_0	R/W	0h	Counter event select inputs for unit 0. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.5 CLB_FSM_EXTRA_IN0 Register (Offset = 14h) [Reset = 00000000h]

CLB_FSM_EXTRA_IN0 is shown in Figure 11-25 and described in Table 11-28.

Return to the Summary Table.

FSM Extra EXT_IN0

Figure 11-25. CLB_FSM_EXTRA_IN0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	SEL_2					SEL_1					SEL_0				
R/W1C-0h																	R/W-0h					R/W-0h					R/W-0h				

Table 11-28. CLB_FSM_EXTRA_IN0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14-10	SEL_2	R/W	0h	FSM block extra external IN0 select inputs for unit 2. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	SEL_1	R/W	0h	FSM block extra external IN0 select inputs for unit 1. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	SEL_0	R/W	0h	FSM block extra external IN0 select inputs for unit 0. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.6 CLB_FSM_EXTERNAL_IN0 Register (Offset = 18h) [Reset = 00000000h]

CLB_FSM_EXTERNAL_IN0 is shown in Figure 11-26 and described in Table 11-29.

Return to the Summary Table.

FSM EXT_IN0

Figure 11-26. CLB_FSM_EXTERNAL_IN0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	SEL_2					SEL_1					SEL_0				
R/W1C-0h																	R/W-0h					R/W-0h					R/W-0h				

Table 11-29. CLB_FSM_EXTERNAL_IN0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14-10	SEL_2	R/W	0h	FSM block EXT_IN0 select input for unit 2. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	SEL_1	R/W	0h	FSM block EXT_IN0 select input for unit 1. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	SEL_0	R/W	0h	FSM block EXT_IN0 select input for unit 0. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.7 CLB_FSM_EXTERNAL_IN1 Register (Offset = 1Ch) [Reset = 00000000h]

CLB_FSM_EXTERNAL_IN1 is shown in Figure 11-27 and described in Table 11-30.

Return to the Summary Table.

FSM_EXT_IN1

Figure 11-27. CLB_FSM_EXTERNAL_IN1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	SEL_2					SEL_1					SEL_0				
R/W1C-0h																	R/W-0h					R/W-0h					R/W-0h				

Table 11-30. CLB_FSM_EXTERNAL_IN1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14-10	SEL_2	R/W	0h	FSM block EXT_IN1 select input for unit 2. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	SEL_1	R/W	0h	FSM block EXT_IN1 select input for unit 1. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	SEL_0	R/W	0h	FSM block EXT_IN1 select input for unit 0. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.8 CLB_FSM_EXTRA_IN1 Register (Offset = 20h) [Reset = 00000000h]

CLB_FSM_EXTRA_IN1 is shown in Figure 11-28 and described in Table 11-31.

Return to the Summary Table.

FSM Extra_EXT_IN1

Figure 11-28. CLB_FSM_EXTRA_IN1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	SEL_2					SEL_1					SEL_0				
R/W1C-0h																	R/W-0h					R/W-0h					R/W-0h				

Table 11-31. CLB_FSM_EXTRA_IN1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14-10	SEL_2	R/W	0h	FSM block extra external IN1 select inputs for unit 2. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	SEL_1	R/W	0h	FSM block extra external IN1 select inputs for unit 1. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	SEL_0	R/W	0h	FSM block extra external IN1 select inputs for unit 0. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.9 CLB_LUT4_IN0 Register (Offset = 24h) [Reset = 00000000h]

CLB_LUT4_IN0 is shown in Figure 11-29 and described in Table 11-32.

Return to the Summary Table.

LUT4_0/1/2 IN0 input source

Figure 11-29. CLB_LUT4_IN0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	SEL_2					SEL_1					SEL_0				
R/W1C-0h																	R/W-0h					R/W-0h					R/W-0h				

Table 11-32. CLB_LUT4_IN0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14-10	SEL_2	R/W	0h	LUT4 block IN0 select inputs for unit 2. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	SEL_1	R/W	0h	LUT4 block IN0 select inputs for unit 1. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	SEL_0	R/W	0h	LUT4 block IN0 select inputs for unit 0. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.10 CLB_LUT4_IN1 Register (Offset = 28h) [Reset = 00000000h]

CLB_LUT4_IN1 is shown in Figure 11-30 and described in Table 11-33.

Return to the Summary Table.

LUT4_0/1/2 IN1 input source

Figure 11-30. CLB_LUT4_IN1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	SEL_2					SEL_1					SEL_0				
R/W1C-0h																	R/W-0h					R/W-0h					R/W-0h				

Table 11-33. CLB_LUT4_IN1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14-10	SEL_2	R/W	0h	LUT4 block IN1 select inputs for unit 2. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	SEL_1	R/W	0h	LUT4 block IN1 select inputs for unit 1. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	SEL_0	R/W	0h	LUT4 block IN1 select inputs for unit 0. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.11 CLB_LUT4_IN2 Register (Offset = 2Ch) [Reset = 00000000h]

CLB_LUT4_IN2 is shown in Figure 11-31 and described in Table 11-34.

Return to the Summary Table.

LUT4_0/1/2 IN2 input source

Figure 11-31. CLB_LUT4_IN2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	SEL_2					SEL_1					SEL_0				
R/W1C-0h																	R/W-0h					R/W-0h					R/W-0h				

Table 11-34. CLB_LUT4_IN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14-10	SEL_2	R/W	0h	LUT4 block IN2 select inputs for unit 2. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	SEL_1	R/W	0h	LUT4 block IN2 select inputs for unit 1. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	SEL_0	R/W	0h	LUT4 block IN2 select inputs for unit 0. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.12 CLB_LUT4_IN3 Register (Offset = 30h) [Reset = 00000000h]

CLB_LUT4_IN3 is shown in Figure 11-32 and described in Table 11-35.

Return to the Summary Table.

LUT4_0/1/2 IN3 input source

Figure 11-32. CLB_LUT4_IN3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	SEL_2					SEL_1					SEL_0				
R/W1C-0h																	R/W-0h					R/W-0h					R/W-0h				

Table 11-35. CLB_LUT4_IN3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14-10	SEL_2	R/W	0h	LUT4 block IN3 select inputs for unit 2. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	SEL_1	R/W	0h	LUT4 block IN3 select inputs for unit 1. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	SEL_0	R/W	0h	LUT4 block IN3 select inputs for unit 0. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.13 CLB_FSM_LUT_FN1_0 Register (Offset = 38h) [Reset = 00000000h]

CLB_FSM_LUT_FN1_0 is shown in Figure 11-33 and described in Table 11-36.

Return to the Summary Table.

LUT function for FSM Unit 1 and Unit 0

Figure 11-33. CLB_FSM_LUT_FN1_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FN1																FN0															
R/W-0h																R/W-0h															

Table 11-36. CLB_FSM_LUT_FN1_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	FN1	R/W	0h	FSM block LUT output function for unit 1 Reset type: SYSRSn
15-0	FN0	R/W	0h	FSM block LUT output function for unit 0 Reset type: SYSRSn

11.10.2.14 CLB_FSM_LUT_FN2 Register (Offset = 3Ch) [Reset = 00000000h]

CLB_FSM_LUT_FN2 is shown in Figure 11-34 and described in Table 11-37.

Return to the Summary Table.

LUT function for FSM Unit 2

Figure 11-34. CLB_FSM_LUT_FN2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FN1															
R/W1C-0h																R/W-0h															

Table 11-37. CLB_FSM_LUT_FN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W1C	0h	Reserved
15-0	FN1	R/W	0h	LUT4 output function for unit 2 Reset type: SYSRSn

11.10.2.15 CLB_LUT4_FN1_0 Register (Offset = 40h) [Reset = 00000000h]

CLB_LUT4_FN1_0 is shown in Figure 11-35 and described in Table 11-38.

Return to the Summary Table.

LUT function for LUT4 block of Unit 1 and 0

Figure 11-35. CLB_LUT4_FN1_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FN1																FN0															
R/W-0h																R/W-0h															

Table 11-38. CLB_LUT4_FN1_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	FN1	R/W	0h	LUT4 output function for unit 1 Reset type: SYSRSn
15-0	FN0	R/W	0h	LUT4 output function for unit 0 Reset type: SYSRSn

11.10.2.16 CLB_LUT4_FN2 Register (Offset = 44h) [Reset = 00000000h]

CLB_LUT4_FN2 is shown in Figure 11-36 and described in Table 11-39.

Return to the Summary Table.

LUT function for LUT4 block of Unit 2

Figure 11-36. CLB_LUT4_FN2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FN1															
R/W1C-0h																R/W-0h															

Table 11-39. CLB_LUT4_FN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W1C	0h	Reserved
15-0	FN1	R/W	0h	LUT4 output function for unit 2 Reset type: SYSRSn

11.10.2.17 CLB_FSM_NEXT_STATE_0 Register (Offset = 48h) [Reset = 00000000h]

CLB_FSM_NEXT_STATE_0 is shown in Figure 11-37 and described in Table 11-40.

Return to the Summary Table.

FSM Next state equations for Unit 0

Figure 11-37. CLB_FSM_NEXT_STATE_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S1																S0															
R/W-0h																R/W-0h															

Table 11-40. CLB_FSM_NEXT_STATE_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	S1	R/W	0h	FSM next state function for S1, unit0 Reset type: SYSRSn
15-0	S0	R/W	0h	FSM next state function for S0, unit0 Reset type: SYSRSn

11.10.2.18 CLB_FSM_NEXT_STATE_1 Register (Offset = 4Ch) [Reset = 00000000h]

CLB_FSM_NEXT_STATE_1 is shown in Figure 11-38 and described in Table 11-41.

Return to the Summary Table.

FSM Next state equations for Unit 1

Figure 11-38. CLB_FSM_NEXT_STATE_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S1																S0															
R/W-0h																R/W-0h															

Table 11-41. CLB_FSM_NEXT_STATE_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	S1	R/W	0h	FSM next state function for S1, unit1 Reset type: SYSRSn
15-0	S0	R/W	0h	FSM next state function for S0, unit1 Reset type: SYSRSn

11.10.2.19 CLB_FSM_NEXT_STATE_2 Register (Offset = 50h) [Reset = 00000000h]

CLB_FSM_NEXT_STATE_2 is shown in Figure 11-39 and described in Table 11-42.

Return to the Summary Table.

FSM Next state equations for Unit 2

Figure 11-39. CLB_FSM_NEXT_STATE_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S1																S0															
R/W-0h																R/W-0h															

Table 11-42. CLB_FSM_NEXT_STATE_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	S1	R/W	0h	FSM next state function for S1, unit2 Reset type: SYSRSn
15-0	S0	R/W	0h	FSM next state function for S0, unit2 Reset type: SYSRSn

11.10.2.20 CLB_MISC_CONTROL Register (Offset = 54h) [Reset = 00000000h]

CLB_MISC_CONTROL is shown in Figure 11-40 and described in Table 11-43.

Return to the Summary Table.

Static controls for Ctr,FSM

Figure 11-40. CLB_MISC_CONTROL Register

31	30	29	28	27	26	25	24
RESERVED					COUNT2_LFSR_EN	COUNT1_LFSR_EN	COUNT0_LFSR_EN
R-0-0h					R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
COUNT2_MATCH2_TAP_EN	COUNT1_MATCH2_TAP_EN	COUNT0_MATCH2_TAP_EN	COUNT2_MATCH1_TAP_EN	COUNT1_MATCH1_TAP_EN	COUNT0_MATCH1_TAP_EN	FSM_EXTRA_SEL1_2	FSM_EXTRA_SEL0_2
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
FSM_EXTRA_SEL1_1	FSM_EXTRA_SEL0_1	FSM_EXTRA_SEL1_0	FSM_EXTRA_SEL0_0	COUNT_SERIALIZER_2	COUNT_SERIALIZER_1	COUNT_SERIALIZER_0	COUNT_EVENT_CTRL_2
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
COUNT_DIR_2	COUNT_ADD_SHIFT_2	COUNT_EVENT_CTRL_1	COUNT_DIR_1	COUNT_ADD_SHIFT_1	COUNT_EVENT_CTRL_0	COUNT_DIR_0	COUNT_ADD_SHIFT_0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 11-43. CLB_MISC_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	Reserved
26	COUNT2_LFSR_EN	R/W	0h	Defines if Counter 2 should operate in LFSR mode. This should be set to 1 only if it is in the SERIALIZER mode. 0 = Selects normal serializer operation 1 = Selects LFSR mode of operation Reset type: SYSRSn
25	COUNT1_LFSR_EN	R/W	0h	Defines if Counter 1 should operate in LFSR mode. This should be set to 1 only if it is in the SERIALIZER mode. 0 = Selects normal serializer operation 1 = Selects LFSR mode of operation Reset type: SYSRSn
24	COUNT0_LFSR_EN	R/W	0h	Defines if Counter 0 should operate in LFSR mode. This should be set to 1 only if it is in the SERIALIZER mode. 0 = Selects normal serializer operation 1 = Selects LFSR mode of operation Reset type: SYSRSn
23	COUNT2_MATCH2_TAP_EN	R/W	0h	Defines if the Match2 output should come from the match unit or tapped from a bit position of the counter 0 = Selects Match2 comaprison output 1 = Selects Bit position defined by Match2_Tap_val Reset type: SYSRSn
22	COUNT1_MATCH2_TAP_EN	R/W	0h	Defines if the Match2 output should come from the match unit or tapped from a bit position of the counter 0 = Selects Match2 comaprison output 1 = Selects Bit position defined by Match2_Tap_val Reset type: SYSRSn

Table 11-43. CLB_MISC_CONTROL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	COUNT0_MATCH2_TAP_EN	R/W	0h	Defines if the Match2 output should come from the match unit or tapped from a bit position of the counter 0 = Selects Match2 comaprison output 1 = Selects Bit position defined by Match2_Tap_val Reset type: SYSRSn
20	COUNT2_MATCH1_TAP_EN	R/W	0h	Defines if the Match1 output should come from the match unit or tapped from a bit position of the counter 0 = Selects Match1 comaprison output 1 = Selects Bit position defined by Match1_Tap_val Reset type: SYSRSn
19	COUNT1_MATCH1_TAP_EN	R/W	0h	Defines if the Match1 output should come from the match unit or tapped from a bit position of the counter 0 = Selects Match1 comaprison output 1 = Selects Bit position defined by Match1_Tap_val Reset type: SYSRSn
18	COUNT0_MATCH1_TAP_EN	R/W	0h	Defines if the Match1 output should come from the match unit or tapped from a bit position of the counter 0 = Selects Match1 comaprison output 1 = Selects Bit position defined by Match1_Tap_val Reset type: SYSRSn
17	FSM_EXTRA_SEL1_2	R/W	0h	Defines which input should be selected for the FSM LUT of UNIT 2 0 = Selects State S1 for the FSM LUT 1 = Selects EXTRA_EXT_IN1 for the FSM LUT Reset type: SYSRSn
16	FSM_EXTRA_SEL0_2	R/W	0h	Defines which input should be selected for the FSM LUT of UNIT 2 0 = Selects State S0 for the FSM LUT 1 = Selects EXTRA_EXT_IN0 for the FSM LUT Reset type: SYSRSn
15	FSM_EXTRA_SEL1_1	R/W	0h	Defines which input should be selected for the FSM LUT of UNIT 1 0 = Selects State S1 for the FSM LUT 1 = Selects EXTRA_EXT_IN1 for the FSM LUT Reset type: SYSRSn
14	FSM_EXTRA_SEL0_1	R/W	0h	Defines which input should be selected for the FSM LUT of UNIT 1 0 = Selects State S0 for the FSM LUT 1 = Selects EXTRA_EXT_IN0 for the FSM LUT Reset type: SYSRSn
13	FSM_EXTRA_SEL1_0	R/W	0h	Defines which input should be selected for the FSM LUT of UNIT 0 0 = Selects State S1 for the FSM LUT 1 = Selects EXTRA_EXT_IN1 for the FSM LUT Reset type: SYSRSn
12	FSM_EXTRA_SEL0_0	R/W	0h	Defines which input should be selected for the FSM LUT of UNIT 0 0 = Selects State S0 for the FSM LUT 1 = Selects EXTRA_EXT_IN0 for the FSM LUT Reset type: SYSRSn
11	COUNT_SERIALIZER_2	R/W	0h	Controls if the Counter of UNIT 2 is the Serialzer mode or not. 0 = Normal mode 1 = Serializer mode Reset type: SYSRSn
10	COUNT_SERIALIZER_1	R/W	0h	Controls if the Counter of UNIT 1 is the Serialzer mode or not. 0 = Normal mode 1 = Serializer mode Reset type: SYSRSn
9	COUNT_SERIALIZER_0	R/W	0h	Controls if the Counter of UNIT 0 is the Serialzer mode or not. 0 = Normal mode 1 = Serializer mode Reset type: SYSRSn

Table 11-43. CLB_MISC_CONTROL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	COUNT_EVENT_CTRL_2	R/W	0h	Controls the actions on an EVENT for UNIT2. Must be 0 for indirect loads and HLC loads of the counter to take effect. 0 = No add or shift, but load the predefined value 1 = Based on other bits, add/shift with the predefined value Reset type: SYSRSn
7	COUNT_DIR_2	R/W	0h	Controls add/shift direction for UNIT 2 0 = right shift or subtract 1 = left shift or add Reset type: SYSRSn
6	COUNT_ADD_SHIFT_2	R/W	0h	Controls whether the UNIT 2 counter will do an ADD or a SHIFT on an EVENT. 0 =Shift 1 = ADD Reset type: SYSRSn
5	COUNT_EVENT_CTRL_1	R/W	0h	Controls the actions on an EVENT for UNIT1. Must be 0 for indirect loads and HLC loads of the counter to take effect. 0 = No add or shift, but load the predefined value 1 = Based on other bits, add/shift with the predefined value Reset type: SYSRSn
4	COUNT_DIR_1	R/W	0h	Controls add/shift direction for UNIT 1 0 = right shift or subtract 1 = left shift or add Reset type: SYSRSn
3	COUNT_ADD_SHIFT_1	R/W	0h	Controls whether the UNIT 1 counter will do an ADD or a SHIFT on an EVENT. 0 = Shift 1 = ADD Reset type: SYSRSn
2	COUNT_EVENT_CTRL_0	R/W	0h	Controls the actions on an EVENT for UNIT1. Must be 0 for indirect loads and HLC loads of the counter to take effect. 0 = No add or shift, but load the predefined value 1 = Based on other bits, add/shift with the predefined value Reset type: SYSRSn
1	COUNT_DIR_0	R/W	0h	Controls add/shift direction for UNIT 0 0 = right shift or subtract 1 = left shift or add Reset type: SYSRSn
0	COUNT_ADD_SHIFT_0	R/W	0h	Controls whether the UNIT 0 counter will do an ADD or a SHIFT on an EVENT. 0 = Shift 1 = ADD Reset type: SYSRSn

11.10.2.21 CLB_OUTPUT_LUT_0 Register (Offset = 58h) [Reset = 00000000h]

CLB_OUTPUT_LUT_0 is shown in Figure 11-41 and described in Table 11-44.

Return to the Summary Table.

Inp Sel, LUT fns for Out0

Figure 11-41. CLB_OUTPUT_LUT_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									FN								IN2					IN1					IN0				
R/W1C-0h									R/W-0h								R/W-0h					R/W-0h					R/W-0h				

Table 11-44. CLB_OUTPUT_LUT_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W1C	0h	Reserved
22-15	FN	R/W	0h	Output function for output LUT Reset type: SYSRSn
14-10	IN2	R/W	0h	Select value for IN2 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	IN1	R/W	0h	Select value for IN1 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	IN0	R/W	0h	Select value for IN0 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.22 CLB_OUTPUT_LUT_1 Register (Offset = 5Ch) [Reset = 00000000h]

CLB_OUTPUT_LUT_1 is shown in Figure 11-42 and described in Table 11-45.

Return to the Summary Table.

Inp Sel, LUT fns for Out1

Figure 11-42. CLB_OUTPUT_LUT_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									FN								IN2					IN1					IN0				
R/W1C-0h									R/W-0h								R/W-0h					R/W-0h					R/W-0h				

Table 11-45. CLB_OUTPUT_LUT_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W1C	0h	Reserved
22-15	FN	R/W	0h	Output function for output LUT Reset type: SYSRSn
14-10	IN2	R/W	0h	Select value for IN2 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	IN1	R/W	0h	Select value for IN1 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	IN0	R/W	0h	Select value for IN0 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.23 CLB_OUTPUT_LUT_2 Register (Offset = 60h) [Reset = 00000000h]

CLB_OUTPUT_LUT_2 is shown in Figure 11-43 and described in Table 11-46.

Return to the Summary Table.

Inp Sel, LUT fns for Out2

Figure 11-43. CLB_OUTPUT_LUT_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									FN								IN2					IN1					IN0				
R/W1C-0h									R/W-0h								R/W-0h					R/W-0h					R/W-0h				

Table 11-46. CLB_OUTPUT_LUT_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W1C	0h	Reserved
22-15	FN	R/W	0h	Output function for output LUT Reset type: SYSRSn
14-10	IN2	R/W	0h	Select value for IN2 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	IN1	R/W	0h	Select value for IN1 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	IN0	R/W	0h	Select value for IN0 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.24 CLB_OUTPUT_LUT_3 Register (Offset = 64h) [Reset = 00000000h]

CLB_OUTPUT_LUT_3 is shown in Figure 11-44 and described in Table 11-47.

Return to the Summary Table.

Inp Sel, LUT fns for Out3

Figure 11-44. CLB_OUTPUT_LUT_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									FN								IN2					IN1					IN0				
R/W1C-0h									R/W-0h								R/W-0h					R/W-0h					R/W-0h				

Table 11-47. CLB_OUTPUT_LUT_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W1C	0h	Reserved
22-15	FN	R/W	0h	Output function for output LUT Reset type: SYSRSn
14-10	IN2	R/W	0h	Select value for IN2 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	IN1	R/W	0h	Select value for IN1 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	IN0	R/W	0h	Select value for IN0 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.25 CLB_OUTPUT_LUT_4 Register (Offset = 68h) [Reset = 00000000h]

CLB_OUTPUT_LUT_4 is shown in Figure 11-45 and described in Table 11-48.

Return to the Summary Table.

Inp Sel, LUT fns for Out4

Figure 11-45. CLB_OUTPUT_LUT_4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									FN								IN2					IN1					IN0				
R/W1C-0h									R/W-0h								R/W-0h					R/W-0h					R/W-0h				

Table 11-48. CLB_OUTPUT_LUT_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W1C	0h	Reserved
22-15	FN	R/W	0h	Output function for output LUT Reset type: SYSRSn
14-10	IN2	R/W	0h	Select value for IN2 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	IN1	R/W	0h	Select value for IN1 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	IN0	R/W	0h	Select value for IN0 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.26 CLB_OUTPUT_LUT_5 Register (Offset = 6Ch) [Reset = 00000000h]

CLB_OUTPUT_LUT_5 is shown in Figure 11-46 and described in Table 11-49.

Return to the Summary Table.

Inp Sel, LUT fns for Out5

Figure 11-46. CLB_OUTPUT_LUT_5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									FN								IN2					IN1					IN0				
R/W1C-0h									R/W-0h								R/W-0h					R/W-0h					R/W-0h				

Table 11-49. CLB_OUTPUT_LUT_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W1C	0h	Reserved
22-15	FN	R/W	0h	Output function for output LUT Reset type: SYSRSn
14-10	IN2	R/W	0h	Select value for IN2 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	IN1	R/W	0h	Select value for IN1 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	IN0	R/W	0h	Select value for IN0 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.27 CLB_OUTPUT_LUT_6 Register (Offset = 70h) [Reset = 00000000h]

CLB_OUTPUT_LUT_6 is shown in Figure 11-47 and described in Table 11-50.

Return to the Summary Table.

Inp Sel, LUT fns for Out6

Figure 11-47. CLB_OUTPUT_LUT_6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									FN								IN2					IN1					IN0				
R/W1C-0h									R/W-0h								R/W-0h					R/W-0h					R/W-0h				

Table 11-50. CLB_OUTPUT_LUT_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W1C	0h	Reserved
22-15	FN	R/W	0h	Output function for output LUT Reset type: SYSRSn
14-10	IN2	R/W	0h	Select value for IN2 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	IN1	R/W	0h	Select value for IN1 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	IN0	R/W	0h	Select value for IN0 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.28 CLB_OUTPUT_LUT_7 Register (Offset = 74h) [Reset = 00000000h]

CLB_OUTPUT_LUT_7 is shown in Figure 11-48 and described in Table 11-51.

Return to the Summary Table.

Inp Sel, LUT fns for Out7

Figure 11-48. CLB_OUTPUT_LUT_7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									FN								IN2					IN1					IN0				
R/W1C-0h									R/W-0h								R/W-0h					R/W-0h					R/W-0h				

Table 11-51. CLB_OUTPUT_LUT_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W1C	0h	Reserved
22-15	FN	R/W	0h	Output function for output LUT Reset type: SYSRSn
14-10	IN2	R/W	0h	Select value for IN2 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	IN1	R/W	0h	Select value for IN1 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	IN0	R/W	0h	Select value for IN0 of output LUT. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.29 CLB_HLC_EVENT_SEL Register (Offset = 78h) [Reset = 00000000h]

CLB_HLC_EVENT_SEL is shown in Figure 11-49 and described in Table 11-52.

Return to the Summary Table.

Event Selector register for the High Level controller

Figure 11-49. CLB_HLC_EVENT_SEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
ALT_EVENT3_SEL	ALT_EVENT2_SEL	ALT_EVENT1_SEL	ALT_EVENT0_SEL	EVENT3_SEL			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h			
15	14	13	12	11	10	9	8
EVENT3_SEL	EVENT2_SEL					EVENT1_SEL	
R/W-0h	R/W-0h					R/W-0h	
7	6	5	4	3	2	1	0
EVENT1_SEL			EVENT0_SEL				
R/W-0h			R/W-0h				

Table 11-52. CLB_HLC_EVENT_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	Reserved
23	ALT_EVENT3_SEL	R/W	0h	Defines selection of alternate inputs for EVENT3 Reset type: SYSRSn
22	ALT_EVENT2_SEL	R/W	0h	Defines selection of alternate inputs for EVENT2 Reset type: SYSRSn
21	ALT_EVENT1_SEL	R/W	0h	Defines selection of alternate inputs for EVENT1 Reset type: SYSRSn
20	ALT_EVENT0_SEL	R/W	0h	Defines selection of alternate inputs for EVENT0 Reset type: SYSRSn
19-15	EVENT3_SEL	R/W	0h	5 bit select value for EVENT3 of the High Level Controller. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
14-10	EVENT2_SEL	R/W	0h	5 bit select value for EVENT2 of the High Level Controller. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
9-5	EVENT1_SEL	R/W	0h	5 bit select value for EVENT1 of the High Level Controller. See the Static Switch Block Output Mux Table. Reset type: SYSRSn
4-0	EVENT0_SEL	R/W	0h	5 bit select value for EVENT0 of the High Level Controller. See the Static Switch Block Output Mux Table. Reset type: SYSRSn

11.10.2.30 CLB_COUNT_MATCH_TAP_SEL Register (Offset = 7Ch) [Reset = 00000000h]

CLB_COUNT_MATCH_TAP_SEL is shown in Figure 11-50 and described in Table 11-53.

Return to the Summary Table.

Counter tap values for match1 and match2 outputs

Figure 11-50. CLB_COUNT_MATCH_TAP_SEL Register

31	30	29	28	27	26	25	24
RESERVED	COUNT2_MATCH2					COUNT1_MATCH2	
R-0-0h	R/W-0h					R/W-0h	
23	22	21	20	19	18	17	16
COUNT1_MATCH2			COUNT0_MATCH2				
R/W-0h			R/W-0h				
15	14	13	12	11	10	9	8
RESERVED	COUNT2_MATCH1					COUNT1_MATCH1	
R-0-0h	R/W-0h					R/W-0h	
7	6	5	4	3	2	1	0
COUNT1_MATCH1			COUNT0_MATCH1				
R/W-0h			R/W-0h				

Table 11-53. CLB_COUNT_MATCH_TAP_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0	0h	Reserved
30-26	COUNT2_MATCH2	R/W	0h	5 bit MUX Select for Match2 Tap for Counter Unit 2 Reset type: SYSRSn
25-21	COUNT1_MATCH2	R/W	0h	5 bit MUX Select for Match2 Tap for Counter Unit 1 Reset type: SYSRSn
20-16	COUNT0_MATCH2	R/W	0h	5 bit MUX Select for Match2 Tap for Counter Unit 0 Reset type: SYSRSn
15	RESERVED	R-0	0h	Reserved
14-10	COUNT2_MATCH1	R/W	0h	5 bit MUX Select for Match1 Tap for Counter Unit 2 Reset type: SYSRSn
9-5	COUNT1_MATCH1	R/W	0h	5 bit MUX Select for Match1 Tap for Counter Unit 1 Reset type: SYSRSn
4-0	COUNT0_MATCH1	R/W	0h	5 bit MUX Select for Match1 Tap for Counter Unit 0 Reset type: SYSRSn

11.10.2.31 CLB_OUTPUT_COND_CTRL_0 Register (Offset = 80h) [Reset = 00000000h]

CLB_OUTPUT_COND_CTRL_0 is shown in Figure 11-51 and described in Table 11-54.

Return to the Summary Table.

Output conditioning control for output 0

Figure 11-51. CLB_OUTPUT_COND_CTRL_0 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W1C-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W1C-0h							
15	14	13	12	11	10	9	8
RESERVED	ASYNC_COND_EN	SEL_RAW_IN	HW_RLS_CTRL_SEL	HW_GATING_CTRL_SEL	SEL_RELEASE_CTRL		
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h		
7	6	5	4	3	2	1	0
SEL_GATING_CTRL			LEVEL_3_SEL		LEVEL_2_SEL		LEVEL_1_SEL
R/W1C-0h			R/W1C-0h		R/W1C-0h		R/W1C-0h

Table 11-54. CLB_OUTPUT_COND_CTRL_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14	ASYNC_COND_EN	R/W1C	0h	Controls whether the output will pass through the asynchronous conditioning block or bypass it. 0 Bypass the asynchronous conditioning block 1 Enable the asynchronous conditioning path Reset type: SYSRSn
13	SEL_RAW_IN	R/W1C	0h	Controls whether the CELL outputs or inputs are sent to the output conditioning block logic. 0 = CELL output (internally delayed by 1 cycle) is used. 1 = CELL input (raw) is used. Reset type: SYSRSn
12	HW_RLS_CTRL_SEL	R/W1C	0h	Controls whether the HW (CELL outputs) or software (GP_REG) will act as the release control 0 SW register value will act as release control 1 Selected CELL output will act as release control Reset type: SYSRSn
11	HW_GATING_CTRL_SEL	R/W1C	0h	Controls whether the HW (CELL outputs) or software (GP_REG) will act as the gating control 0 SW register value will act as gating control 1 Selected CELL output will act as gating control Reset type: SYSRSn
10-8	SEL_RELEASE_CTRL	R/W1C	0h	3 bit MUX selects which will select one of the 8 CELL outputs for Release control. Reset type: SYSRSn
7-5	SEL_GATING_CTRL	R/W1C	0h	3 bit MUX selects which will select one of the 8 CELL outputs for Gating control. Reset type: SYSRSn

Table 11-54. CLB_OUTPUT_COND_CTRL_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-3	LEVEL_3_SEL	R/W1C	0h	Controls Third level Mux select 00 Input Signal will be sent as is to the output 01 Rising edge of Input signal will cause asynchronous CLEAR of the output 10 Rising edge of Input signal will cause asynchronous SET of the output 11 Input Signal delayed by 1 clock cycle will be sent to the output Reset type: SYSRSn
2-1	LEVEL_2_SEL	R/W1C	0h	Controls Second level Mux select 00 Input Signal sent as output to next level 01 Input Signal AND Gating_control sent as output to next level 10 Input Signal OR Gating_control sent as output to next level 11 Input Signal XOR Gating_control sent as output to next level Reset type: SYSRSn
0	LEVEL_1_SEL	R/W1C	0h	First level MUX select value 0 Direct signal sent as output to next level 1 Inverted signal sent as output to the next level Reset type: SYSRSn

11.10.2.32 CLB_OUTPUT_COND_CTRL_1 Register (Offset = 84h) [Reset = 00000000h]

CLB_OUTPUT_COND_CTRL_1 is shown in Figure 11-52 and described in Table 11-55.

Return to the Summary Table.

Output conditioning control for output 1

Figure 11-52. CLB_OUTPUT_COND_CTRL_1 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W1C-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W1C-0h							
15	14	13	12	11	10	9	8
RESERVED	ASYNC_COND_EN	SEL_RAW_IN	HW_RLS_CTRL_SEL	HW_GATING_CTRL_SEL	SEL_RELEASE_CTRL		
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h		
7	6	5	4	3	2	1	0
SEL_GATING_CTRL			LEVEL_3_SEL		LEVEL_2_SEL		LEVEL_1_SEL
R/W1C-0h			R/W1C-0h		R/W1C-0h		R/W1C-0h

Table 11-55. CLB_OUTPUT_COND_CTRL_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14	ASYNC_COND_EN	R/W1C	0h	Controls whether the output will pass through the asynchronous conditioning block or bypass it. 0 Bypass the asynchronous conditioning block 1 Enable the asynchronous conditioning path Reset type: SYSRSn
13	SEL_RAW_IN	R/W1C	0h	Controls whether the CELL outputs or inputs are sent to the output conditioning block logic. 0 = CELL output (internally delayed by 1 cycle) is used. 1 = CELL input (raw) is used. Reset type: SYSRSn
12	HW_RLS_CTRL_SEL	R/W1C	0h	Controls whether the HW (CELL outputs) or software (GP_REG) will act as the release control 0 SW register value will act as release control 1 Selected CELL output will act as release control Reset type: SYSRSn
11	HW_GATING_CTRL_SEL	R/W1C	0h	Controls whether the HW (CELL outputs) or software (GP_REG) will act as the gating control 0 SW register value will act as gating control 1 Selected CELL output will act as gating control Reset type: SYSRSn
10-8	SEL_RELEASE_CTRL	R/W1C	0h	3 bit MUX selects which will select one of the 8 CELL outputs for Release control. Reset type: SYSRSn
7-5	SEL_GATING_CTRL	R/W1C	0h	3 bit MUX selects which will select one of the 8 CELL outputs for Gating control. Reset type: SYSRSn

Table 11-55. CLB_OUTPUT_COND_CTRL_1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-3	LEVEL_3_SEL	R/W1C	0h	Controls Third level Mux select 00 Input Signal will be sent as is to the output 01 Rising edge of Input signal will cause asynchronous CLEAR of the output 10 Rising edge of Input signal will cause asynchronous SET of the output 11 Input Signal delayed by 1 clock cycle will be sent to the output Reset type: SYSRSn
2-1	LEVEL_2_SEL	R/W1C	0h	Controls Second level Mux select 00 Input Signal sent as output to next level 01 Input Signal AND Gating_control sent as output to next level 10 Input Signal OR Gating_control sent as output to next level 11 Input Signal XOR Gating_control sent as output to next level Reset type: SYSRSn
0	LEVEL_1_SEL	R/W1C	0h	First level MUX select value 0 Direct signal sent as output to next level 1 Inverted signal sent as output to the next level Reset type: SYSRSn

11.10.2.33 CLB_OUTPUT_COND_CTRL_2 Register (Offset = 88h) [Reset = 00000000h]

CLB_OUTPUT_COND_CTRL_2 is shown in Figure 11-53 and described in Table 11-56.

Return to the Summary Table.

Output conditioning control for output 2

Figure 11-53. CLB_OUTPUT_COND_CTRL_2 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W1C-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W1C-0h							
15	14	13	12	11	10	9	8
RESERVED	ASYNC_COND_EN	SEL_RAW_IN	HW_RLS_CTRL_SEL	HW_GATING_CTRL_SEL	SEL_RELEASE_CTRL		
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h		
7	6	5	4	3	2	1	0
SEL_GATING_CTRL			LEVEL_3_SEL		LEVEL_2_SEL		LEVEL_1_SEL
R/W1C-0h			R/W1C-0h		R/W1C-0h		R/W1C-0h

Table 11-56. CLB_OUTPUT_COND_CTRL_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14	ASYNC_COND_EN	R/W1C	0h	Controls whether the output will pass through the asynchronous conditioning block or bypass it. 0 Bypass the asynchronous conditioning block 1 Enable the asynchronous conditioning path Reset type: SYSRSn
13	SEL_RAW_IN	R/W1C	0h	Controls whether the CELL outputs or inputs are sent to the output conditioning block logic. 0 = CELL output (internally delayed by 1 cycle) is used. 1 = CELL input (raw) is used. Reset type: SYSRSn
12	HW_RLS_CTRL_SEL	R/W1C	0h	Controls whether the HW (CELL outputs) or software (GP_REG) will act as the release control 0 SW register value will act as release control 1 Selected CELL output will act as release control Reset type: SYSRSn
11	HW_GATING_CTRL_SEL	R/W1C	0h	Controls whether the HW (CELL outputs) or software (GP_REG) will act as the gating control 0 SW register value will act as gating control 1 Selected CELL output will act as gating control Reset type: SYSRSn
10-8	SEL_RELEASE_CTRL	R/W1C	0h	3 bit MUX selects which will select one of the 8 CELL outputs for Release control. Reset type: SYSRSn
7-5	SEL_GATING_CTRL	R/W1C	0h	3 bit MUX selects which will select one of the 8 CELL outputs for Gating control. Reset type: SYSRSn

Table 11-56. CLB_OUTPUT_COND_CTRL_2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-3	LEVEL_3_SEL	R/W1C	0h	Controls Third level Mux select 00 Input Signal will be sent as is to the output 01 Rising edge of Input signal will cause asynchronous CLEAR of the output 10 Rising edge of Input signal will cause asynchronous SET of the output 11 Input Signal delayed by 1 clock cycle will be sent to the output Reset type: SYSRSn
2-1	LEVEL_2_SEL	R/W1C	0h	Controls Second level Mux select 00 Input Signal sent as output to next level 01 Input Signal AND Gating_control sent as output to next level 10 Input Signal OR Gating_control sent as output to next level 11 Input Signal XOR Gating_control sent as output to next level Reset type: SYSRSn
0	LEVEL_1_SEL	R/W1C	0h	First level MUX select value 0 Direct signal sent as output to next level 1 Inverted signal sent as output to the next level Reset type: SYSRSn

11.10.2.34 CLB_OUTPUT_COND_CTRL_3 Register (Offset = 8Ch) [Reset = 00000000h]

CLB_OUTPUT_COND_CTRL_3 is shown in Figure 11-54 and described in Table 11-57.

Return to the Summary Table.

Output conditioning control for output 3

Figure 11-54. CLB_OUTPUT_COND_CTRL_3 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W1C-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W1C-0h							
15	14	13	12	11	10	9	8
RESERVED	ASYNC_COND_EN	SEL_RAW_IN	HW_RLS_CTRL_SEL	HW_GATING_CTRL_SEL	SEL_RELEASE_CTRL		
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h		
7	6	5	4	3	2	1	0
SEL_GATING_CTRL			LEVEL_3_SEL		LEVEL_2_SEL		LEVEL_1_SEL
R/W1C-0h			R/W1C-0h		R/W1C-0h		R/W1C-0h

Table 11-57. CLB_OUTPUT_COND_CTRL_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14	ASYNC_COND_EN	R/W1C	0h	Controls whether the output will pass through the asynchronous conditioning block or bypass it. 0 Bypass the asynchronous conditioning block 1 Enable the asynchronous conditioning path Reset type: SYSRSn
13	SEL_RAW_IN	R/W1C	0h	Controls whether the CELL outputs or inputs are sent to the output conditioning block logic. 0 = CELL output (internally delayed by 1 cycle) is used. 1 = CELL input (raw) is used. Reset type: SYSRSn
12	HW_RLS_CTRL_SEL	R/W1C	0h	Controls whether the HW (CELL outputs) or software (GP_REG) will act as the release control 0 SW register value will act as release control 1 Selected CELL output will act as release control Reset type: SYSRSn
11	HW_GATING_CTRL_SEL	R/W1C	0h	Controls whether the HW (CELL outputs) or software (GP_REG) will act as the gating control 0 SW register value will act as gating control 1 Selected CELL output will act as gating control Reset type: SYSRSn
10-8	SEL_RELEASE_CTRL	R/W1C	0h	3 bit MUX selects which will select one of the 8 CELL outputs for Release control. Reset type: SYSRSn
7-5	SEL_GATING_CTRL	R/W1C	0h	3 bit MUX selects which will select one of the 8 CELL outputs for Gating control. Reset type: SYSRSn

Table 11-57. CLB_OUTPUT_COND_CTRL_3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-3	LEVEL_3_SEL	R/W1C	0h	Controls Third level Mux select 00 Input Signal will be sent as is to the output 01 Rising edge of Input signal will cause asynchronous CLEAR of the output 10 Rising edge of Input signal will cause asynchronous SET of the output 11 Input Signal delayed by 1 clock cycle will be sent to the output Reset type: SYSRSn
2-1	LEVEL_2_SEL	R/W1C	0h	Controls Second level Mux select 00 Input Signal sent as output to next level 01 Input Signal AND Gating_control sent as output to next level 10 Input Signal OR Gating_control sent as output to next level 11 Input Signal XOR Gating_control sent as output to next level Reset type: SYSRSn
0	LEVEL_1_SEL	R/W1C	0h	First level MUX select value 0 Direct signal sent as output to next level 1 Inverted signal sent as output to the next level Reset type: SYSRSn

11.10.2.35 CLB_OUTPUT_COND_CTRL_4 Register (Offset = 90h) [Reset = 00000000h]

CLB_OUTPUT_COND_CTRL_4 is shown in Figure 11-55 and described in Table 11-58.

Return to the Summary Table.

Output conditioning control for output 4

Figure 11-55. CLB_OUTPUT_COND_CTRL_4 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W1C-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W1C-0h							
15	14	13	12	11	10	9	8
RESERVED	ASYNC_COND_EN	SEL_RAW_IN	HW_RLS_CTRL_SEL	HW_GATING_CTRL_SEL	SEL_RELEASE_CTRL		
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h		
7	6	5	4	3	2	1	0
SEL_GATING_CTRL			LEVEL_3_SEL		LEVEL_2_SEL		LEVEL_1_SEL
R/W1C-0h			R/W1C-0h		R/W1C-0h		R/W1C-0h

Table 11-58. CLB_OUTPUT_COND_CTRL_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14	ASYNC_COND_EN	R/W1C	0h	Controls whether the output will pass through the asynchronous conditioning block or bypass it. 0 Bypass the asynchronous conditioning block 1 Enable the asynchronous conditioning path Reset type: SYSRSn
13	SEL_RAW_IN	R/W1C	0h	Controls whether the CELL outputs or inputs are sent to the output conditioning block logic. 0 = CELL output (internally delayed by 1 cycle) is used. 1 = CELL input (raw) is used. Reset type: SYSRSn
12	HW_RLS_CTRL_SEL	R/W1C	0h	Controls whether the HW (CELL outputs) or software (GP_REG) will act as the release control 0 SW register value will act as release control 1 Selected CELL output will act as release control Reset type: SYSRSn
11	HW_GATING_CTRL_SEL	R/W1C	0h	Controls whether the HW (CELL outputs) or software (GP_REG) will act as the gating control 0 SW register value will act as gating control 1 Selected CELL output will act as gating control Reset type: SYSRSn
10-8	SEL_RELEASE_CTRL	R/W1C	0h	3 bit MUX selects which will select one of the 8 CELL outputs for Release control. Reset type: SYSRSn
7-5	SEL_GATING_CTRL	R/W1C	0h	3 bit MUX selects which will select one of the 8 CELL outputs for Gating control. Reset type: SYSRSn

Table 11-58. CLB_OUTPUT_COND_CTRL_4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-3	LEVEL_3_SEL	R/W1C	0h	Controls Third level Mux select 00 Input Signal will be sent as is to the output 01 Rising edge of Input signal will cause asynchronous CLEAR of the output 10 Rising edge of Input signal will cause asynchronous SET of the output 11 Input Signal delayed by 1 clock cycle will be sent to the output Reset type: SYSRSn
2-1	LEVEL_2_SEL	R/W1C	0h	Controls Second level Mux select 00 Input Signal sent as output to next level 01 Input Signal AND Gating_control sent as output to next level 10 Input Signal OR Gating_control sent as output to next level 11 Input Signal XOR Gating_control sent as output to next level Reset type: SYSRSn
0	LEVEL_1_SEL	R/W1C	0h	First level MUX select value 0 Direct signal sent as output to next level 1 Inverted signal sent as output to the next level Reset type: SYSRSn

11.10.2.36 CLB_OUTPUT_COND_CTRL_5 Register (Offset = 94h) [Reset = 00000000h]

CLB_OUTPUT_COND_CTRL_5 is shown in Figure 11-56 and described in Table 11-59.

Return to the Summary Table.

Output conditioning control for output 5

Figure 11-56. CLB_OUTPUT_COND_CTRL_5 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W1C-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W1C-0h							
15	14	13	12	11	10	9	8
RESERVED	ASYNC_COND_EN	SEL_RAW_IN	HW_RLS_CTRL_SEL	HW_GATING_CTRL_SEL	SEL_RELEASE_CTRL		
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h		
7	6	5	4	3	2	1	0
SEL_GATING_CTRL			LEVEL_3_SEL		LEVEL_2_SEL		LEVEL_1_SEL
R/W1C-0h			R/W1C-0h		R/W1C-0h		R/W1C-0h

Table 11-59. CLB_OUTPUT_COND_CTRL_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14	ASYNC_COND_EN	R/W1C	0h	Controls whether the output will pass through the asynchronous conditioning block or bypass it. 0 Bypass the asynchronous conditioning block 1 Enable the asynchronous conditioning path Reset type: SYSRSn
13	SEL_RAW_IN	R/W1C	0h	Controls whether the CELL outputs or inputs are sent to the output conditioning block logic. 0 = CELL output (internally delayed by 1 cycle) is used. 1 = CELL input (raw) is used. Reset type: SYSRSn
12	HW_RLS_CTRL_SEL	R/W1C	0h	Controls whether the HW (CELL outputs) or software (GP_REG) will act as the release control 0 SW register value will act as release control 1 Selected CELL output will act as release control Reset type: SYSRSn
11	HW_GATING_CTRL_SEL	R/W1C	0h	Controls whether the HW (CELL outputs) or software (GP_REG) will act as the gating control 0 SW register value will act as gating control 1 Selected CELL output will act as gating control Reset type: SYSRSn
10-8	SEL_RELEASE_CTRL	R/W1C	0h	3 bit MUX selects which will select one of the 8 CELL outputs for Release control. Reset type: SYSRSn
7-5	SEL_GATING_CTRL	R/W1C	0h	3 bit MUX selects which will select one of the 8 CELL outputs for Gating control. Reset type: SYSRSn

Table 11-59. CLB_OUTPUT_COND_CTRL_5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-3	LEVEL_3_SEL	R/W1C	0h	Controls Third level Mux select 00 Input Signal will be sent as is to the output 01 Rising edge of Input signal will cause asynchronous CLEAR of the output 10 Rising edge of Input signal will cause asynchronous SET of the output 11 Input Signal delayed by 1 clock cycle will be sent to the output Reset type: SYSRSn
2-1	LEVEL_2_SEL	R/W1C	0h	Controls Second level Mux select 00 Input Signal sent as output to next level 01 Input Signal AND Gating_control sent as output to next level 10 Input Signal OR Gating_control sent as output to next level 11 Input Signal XOR Gating_control sent as output to next level Reset type: SYSRSn
0	LEVEL_1_SEL	R/W1C	0h	First level MUX select value 0 Direct signal sent as output to next level 1 Inverted signal sent as output to the next level Reset type: SYSRSn

11.10.2.37 CLB_OUTPUT_COND_CTRL_6 Register (Offset = 98h) [Reset = 00000000h]

CLB_OUTPUT_COND_CTRL_6 is shown in Figure 11-57 and described in Table 11-60.

Return to the Summary Table.

Output conditioning control for output 6

Figure 11-57. CLB_OUTPUT_COND_CTRL_6 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W1C-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W1C-0h							
15	14	13	12	11	10	9	8
RESERVED	ASYNC_COND_EN	SEL_RAW_IN	HW_RLS_CTRL_SEL	HW_GATING_CTRL_SEL	SEL_RELEASE_CTRL		
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h		
7	6	5	4	3	2	1	0
SEL_GATING_CTRL			LEVEL_3_SEL		LEVEL_2_SEL		LEVEL_1_SEL
R/W1C-0h			R/W1C-0h		R/W1C-0h		R/W1C-0h

Table 11-60. CLB_OUTPUT_COND_CTRL_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14	ASYNC_COND_EN	R/W1C	0h	Controls whether the output will pass through the asynchronous conditioning block or bypass it. 0 Bypass the asynchronous conditioning block 1 Enable the asynchronous conditioning path Reset type: SYSRSn
13	SEL_RAW_IN	R/W1C	0h	Controls whether the CELL outputs or inputs are sent to the output conditioning block logic. 0 = CELL output (internally delayed by 1 cycle) is used. 1 = CELL input (raw) is used. Reset type: SYSRSn
12	HW_RLS_CTRL_SEL	R/W1C	0h	Controls whether the HW (CELL outputs) or software (GP_REG) will act as the release control 0 SW register value will act as release control 1 Selected CELL output will act as release control Reset type: SYSRSn
11	HW_GATING_CTRL_SEL	R/W1C	0h	Controls whether the HW (CELL outputs) or software (GP_REG) will act as the gating control 0 SW register value will act as gating control 1 Selected CELL output will act as gating control Reset type: SYSRSn
10-8	SEL_RELEASE_CTRL	R/W1C	0h	3 bit MUX selects which will select one of the 8 CELL outputs for Release control. Reset type: SYSRSn
7-5	SEL_GATING_CTRL	R/W1C	0h	3 bit MUX selects which will select one of the 8 CELL outputs for Gating control. Reset type: SYSRSn

Table 11-60. CLB_OUTPUT_COND_CTRL_6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-3	LEVEL_3_SEL	R/W1C	0h	Controls Third level Mux select 00 Input Signal will be sent as is to the output 01 Rising edge of Input signal will cause asynchronous CLEAR of the output 10 Rising edge of Input signal will cause asynchronous SET of the output 11 Input Signal delayed by 1 clock cycle will be sent to the output Reset type: SYSRSn
2-1	LEVEL_2_SEL	R/W1C	0h	Controls Second level Mux select 00 Input Signal sent as output to next level 01 Input Signal AND Gating_control sent as output to next level 10 Input Signal OR Gating_control sent as output to next level 11 Input Signal XOR Gating_control sent as output to next level Reset type: SYSRSn
0	LEVEL_1_SEL	R/W1C	0h	First level MUX select value 0 Direct signal sent as output to next level 1 Inverted signal sent as output to the next level Reset type: SYSRSn

11.10.2.38 CLB_OUTPUT_COND_CTRL_7 Register (Offset = 9Ch) [Reset = 00000000h]

CLB_OUTPUT_COND_CTRL_7 is shown in Figure 11-58 and described in Table 11-61.

Return to the Summary Table.

Output conditioning control for output 7

Figure 11-58. CLB_OUTPUT_COND_CTRL_7 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W1C-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W1C-0h							
15	14	13	12	11	10	9	8
RESERVED	ASYNC_COND_EN	SEL_RAW_IN	HW_RLS_CTRL_SEL	HW_GATING_CTRL_SEL	SEL_RELEASE_CTRL		
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h		
7	6	5	4	3	2	1	0
SEL_GATING_CTRL			LEVEL_3_SEL		LEVEL_2_SEL		LEVEL_1_SEL
R/W1C-0h			R/W1C-0h		R/W1C-0h		R/W1C-0h

Table 11-61. CLB_OUTPUT_COND_CTRL_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W1C	0h	Reserved
14	ASYNC_COND_EN	R/W1C	0h	Controls whether the output will pass through the asynchronous conditioning block or bypass it. 0 Bypass the asynchronous conditioning block 1 Enable the asynchronous conditioning path Reset type: SYSRSn
13	SEL_RAW_IN	R/W1C	0h	Controls whether the CELL outputs or inputs are sent to the output conditioning block logic. 0 = CELL output (internally delayed by 1 cycle) is used. 1 = CELL input (raw) is used. Reset type: SYSRSn
12	HW_RLS_CTRL_SEL	R/W1C	0h	Controls whether the HW (CELL outputs) or software (GP_REG) will act as the release control 0 SW register value will act as release control 1 Selected CELL output will act as release control Reset type: SYSRSn
11	HW_GATING_CTRL_SEL	R/W1C	0h	Controls whether the HW (CELL outputs) or software (GP_REG) will act as the gating control 0 SW register value will act as gating control 1 Selected CELL output will act as gating control Reset type: SYSRSn
10-8	SEL_RELEASE_CTRL	R/W1C	0h	3 bit MUX selects which will select one of the 8 CELL outputs for Release control. Reset type: SYSRSn
7-5	SEL_GATING_CTRL	R/W1C	0h	3 bit MUX selects which will select one of the 8 CELL outputs for Gating control. Reset type: SYSRSn

Table 11-61. CLB_OUTPUT_COND_CTRL_7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-3	LEVEL_3_SEL	R/W1C	0h	Controls Third level Mux select 00 Input Signal will be sent as is to the output 01 Rising edge of Input signal will cause asynchronous CLEAR of the output 10 Rising edge of Input signal will cause asynchronous SET of the output 11 Input Signal delayed by 1 clock cycle will be sent to the output Reset type: SYSRSn
2-1	LEVEL_2_SEL	R/W1C	0h	Controls Second level Mux select 00 Input Signal sent as output to next level 01 Input Signal AND Gating_control sent as output to next level 10 Input Signal OR Gating_control sent as output to next level 11 Input Signal XOR Gating_control sent as output to next level Reset type: SYSRSn
0	LEVEL_1_SEL	R/W1C	0h	First level MUX select value 0 Direct signal sent as output to next level 1 Inverted signal sent as output to the next level Reset type: SYSRSn

11.10.2.39 CLB_MISC_ACCESS_CTRL Register (Offset = A0h) [Reset = 0000h]

CLB_MISC_ACCESS_CTRL is shown in Figure 11-59 and described in Table 11-62.

Return to the Summary Table.

Miscellaneous Access and enable control

Figure 11-59. CLB_MISC_ACCESS_CTRL Register

15	14	13	12	11	10	9	8
RESERVED							
R/W1C-0h							
7	6	5	4	3	2	1	0
RESERVED						BLKEN	SPIEN
R/W1C-0h						R/W1C-0h	R/W1C-0h

Table 11-62. CLB_MISC_ACCESS_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-2	RESERVED	R/W1C	0h	Reserved
1	BLKEN	R/W1C	0h	This bit is used to block writes to CLB_OUT_EN 0 Writes to CLB_OUT_EN are allowed 1 Writes to CLB_OUT_EN are blocked Reset type: SYSRSn
0	SPIEN	R/W1C	0h	This bit indicates the status of the SPI buffers ability to export CLB output data. 0 Feature Disabled 1 Feature Enabled Reset type: SYSRSn

11.10.2.40 CLB_SPI_DATA_CTRL_HI Register (Offset = A2h) [Reset = 0000h]

CLB_SPI_DATA_CTRL_HI is shown in Figure 11-60 and described in Table 11-63.

Return to the Summary Table.

CLB to SPI buffer control High

Figure 11-60. CLB_SPI_DATA_CTRL_HI Register

15	14	13	12	11	10	9	8
RESERVED			SHIFT				
R/W1C-0h			R/W1C-0h				
7	6	5	4	3	2	1	0
STRB_DEL	RESERVED		STRB				
R/W1C-0h	R/W1C-0h		R/W1C-0h				

Table 11-63. CLB_SPI_DATA_CTRL_HI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R/W1C	0h	Reserved
12-8	SHIFT	R/W1C	0h	This is a 5 bit value which denotes the first bit position of register R0 from which the Least Significant Bit of the output data should start. 00000 Output data is R0[15:0] 00001 Output data is R0[16:1] 00010 Output data is R0[17:2] 00011 Output data is R0[18:3] ... ... 10000 Output data is R0[31:16] Reset type: SYSRSn
7	STRB_DEL	R/W1C	0h	0 Selected strobe event goes directly to SPI module 1 Selected strobe event is delayed by 4 CLB clock cycles to SPI module (This is done to facilitate use of STROBE as the same event to initiate a HLC task that can get the required data on to R0 for SPI data transfer before Strobe becomes valid at SPI block) Reset type: SYSRSn
6-5	RESERVED	R/W1C	0h	Reserved
4-0	STRB	R/W1C	0h	This is a 5 bit value which selects one of the HLC_EVENT inputs to be treated as the data_valid strobe Reset type: SYSRSn

11.10.3 CLB_LOGIC_CONTROL_REGS Registers

Table 11-64 lists the memory-mapped registers for the CLB_LOGIC_CONTROL_REGS registers. All register offset addresses not listed in Table 11-64 should be considered as reserved locations and the register contents should not be modified.

Table 11-64. CLB_LOGIC_CONTROL_REGS Registers

Offset	Acronym	Register Name	Protection
0h	CLB_LOAD_EN	Global enable & indirect load enable control, only Global Enable Bit is LOCK protected	LOCK
4h	CLB_LOAD_ADDR	Indirect address	
8h	CLB_LOAD_DATA	Data for indirect loads	
Ch	CLB_INPUT_FILTER	Input filter selection for both edge detection and synchronizers	LOCK
10h	CLB_IN_MUX_SEL_0	Input selection to decide between Signals and GP register	LOCK
14h	CLB_LCL_MUX_SEL_1	Input Mux selection for local mux	LOCK
18h	CLB_LCL_MUX_SEL_2	Input Mux selection for local mux	LOCK
1Ch	CLB_BUF_PTR	PUSH and PULL pointers	
20h	CLB_GP_REG	General purpose register for CELL inputs	
24h	CLB_OUT_EN	CELL output enable register	
28h	CLB_GLBL_MUX_SEL_1	Global Mux select for CELL inputs	LOCK
2Ch	CLB_GLBL_MUX_SEL_2	Global Mux select for CELL inputs	LOCK
30h	CLB_PRESCALE_CTRL	Prescaler register control	LOCK
40h	CLB_INTR_TAG_REG	Interrupt Tag register	
44h	CLB_LOCK	Lock control register	
48h	CLB_HLC_INSTR_READ_PTR	HLC instruction read pointer	
4Ch	CLB_HLC_INSTR_VALUE	HLC instruction read value	
5Ch	CLB_DBG_OUT_2	Visibility for CLB inputs and final asynchronous outputs	
60h	CLB_DBG_R0	R0 of High level Controller	
64h	CLB_DBG_R1	R1 of High level Controller	
68h	CLB_DBG_R2	R2 of High level Controller	
6Ch	CLB_DBG_R3	R3 of High level Controller	
70h	CLB_DBG_C0	Count of Unit 0	
74h	CLB_DBG_C1	Count of Unit 1	
78h	CLB_DBG_C2	Count of Unit 2	
7Ch	CLB_DBG_OUT	Outputs of various units in the Cell	

Complex bit access types are encoded to fit into small table cells. Table 11-65 shows the codes that are used for access types in this section.

Table 11-65. CLB_LOGIC_CONTROL_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
R-1	R -1	Read Returns 1s
Write Type		
W	W	Write

Table 11-65. CLB_LOGIC_CONTROL_REGS Access Type Codes (continued)

Access Type	Code	Description
W1C	W 1C	Write 1 to clear
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

11.10.3.1 CLB_LOAD_EN Register (Offset = 0h) [Reset = 0000h]

CLB_LOAD_EN is shown in Figure 11-61 and described in Table 11-66.

Return to the Summary Table.

Global enable & indirect load enable control, only Global Enable Bit is LOCK protected

Figure 11-61. CLB_LOAD_EN Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED			PIPELINE_EN	NMI_EN	STOP	GLOBAL_EN	LOAD_EN
R-0-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 11-66. CLB_LOAD_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R-0	0h	Reserved
4	PIPELINE_EN	R/W	0h	This bit controls pipelining of all the CLB operations in HLC and Counter blocks. Pipelined operation is enabled when this bit is set to 1. Reset type: SYSRSn
3	NMI_EN	R/W	0h	This bit controls the generation of NMI along with the interrupt whenever a INTR operation is executed by the HLC. NMI generation is disabled by default. It will be enabled when this bit is set to 1. Reset type: SYSRSn
2	STOP	R/W	0h	This bit defines the behaviour of the sequential elements in the CELL during debug HALTs of the CPU. If this bit is set to 0, the debug HALT condition is ignored. Reset type: SYSRSn
1	GLOBAL_EN	R/W	0h	This bit is a global enable signal for the logic in the CELL. This also acts as a soft reset for the CELL logic. CLB outputs (including LUTs and OUTLUTs) will be gated when this bit is cleared from 1 to 0, i.e., the CLB outputs will be low when GLOBAL_EN is low. Additionally, the FSM and AOC blocks will also be reset. Note that when this bit goes low, the COUNTER blocks and HLC are simply halted, but they will NOT be reset internally. This allows the ability to preload these submodules when GLOBAL_EN is 0. This bit is normally set after all the other configuration settings are completed. This bit is LOCK protected. Reset type: SYSRSn
0	LOAD_EN	R/W	0h	A write with this bit set to 1 will pulse the Load Enable signal for the indirect register loads in the CELL. Reset type: SYSRSn

11.10.3.2 CLB_LOAD_ADDR Register (Offset = 4h) [Reset = 00000000h]

CLB_LOAD_ADDR is shown in Figure 11-62 and described in Table 11-67.

Return to the Summary Table.

Indirect address

Figure 11-62. CLB_LOAD_ADDR Register

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

Table 11-67. CLB_LOAD_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5-0	ADDR	R/W	0h	These are the address bits used for writing to the indirect address space of the CELL. Reset type: SYSRSn

11.10.3.3 CLB_LOAD_DATA Register (Offset = 8h) [Reset = 00000000h]

CLB_LOAD_DATA is shown in Figure 11-63 and described in Table 11-68.

Return to the Summary Table.

Data for indirect loads

Figure 11-63. CLB_LOAD_DATA Register

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

Table 11-68. CLB_LOAD_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/W	0h	This register holds the 32-bit data for writing to the indirect address space of the CELL. Reset type: SYSRSn

11.10.3.4 CLB_INPUT_FILTER Register (Offset = Ch) [Reset = 00000000h]

CLB_INPUT_FILTER is shown in Figure 11-64 and described in Table 11-69.

Return to the Summary Table.

Input filter selection for both edge detection and synchronizers

Figure 11-64. CLB_INPUT_FILTER Register

31	30	29	28	27	26	25	24
PIPE7	PIPE6	PIPE5	PIPE4	PIPE3	PIPE2	PIPE1	PIPE0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
SYNC7	SYNC6	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
FIN7		FIN6		FIN5		FIN4	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
FIN3		FIN2		FIN1		FIN0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 11-69. CLB_INPUT_FILTER Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PIPE7	R/W	0h	Enable pipelining for Input 7 Reset type: SYSRSn
30	PIPE6	R/W	0h	Enable pipelining for Input 6 Reset type: SYSRSn
29	PIPE5	R/W	0h	Enable pipelining for Input 5 Reset type: SYSRSn
28	PIPE4	R/W	0h	Enable pipelining for Input 4 Reset type: SYSRSn
27	PIPE3	R/W	0h	Enable pipelining for Input 3 Reset type: SYSRSn
26	PIPE2	R/W	0h	Enable pipelining for Input 2 Reset type: SYSRSn
25	PIPE1	R/W	0h	Enable pipelining for Input 1 Reset type: SYSRSn
24	PIPE0	R/W	0h	Enable pipelining for Input 0 Reset type: SYSRSn
23	SYNC7	R/W	0h	Synchronizer Select Control for Input 7 Reset type: SYSRSn
22	SYNC6	R/W	0h	Synchronizer Select Control for Input 6 Reset type: SYSRSn
21	SYNC5	R/W	0h	Synchronizer Select Control for Input 5 Reset type: SYSRSn
20	SYNC4	R/W	0h	Synchronizer Select Control for Input 4 Reset type: SYSRSn
19	SYNC3	R/W	0h	Synchronizer Select Control for Input 3 Reset type: SYSRSn

Table 11-69. CLB_INPUT_FILTER Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	SYNC2	R/W	0h	Synchronizer Select Control for Input 2 Reset type: SYSRSn
17	SYNC1	R/W	0h	Synchronizer Select Control for Input 1 Reset type: SYSRSn
16	SYNC0	R/W	0h	Synchronizer Select Control for Input 0 Reset type: SYSRSn
15-14	FIN7	R/W	0h	Input filter selection for CELL Input 7 2 bits are used to define the edge filtering . 00 : No filtering 01 : Rising edge detect 10 : Falling edge detect 11 : Any edge detect Reset type: SYSRSn
13-12	FIN6	R/W	0h	Input filter selection for CELL Input 6 2 bits are used to define the edge filtering . 00 : No filtering 01 : Rising edge detect 10 : Falling edge detect 11 : Any edge detect Reset type: SYSRSn
11-10	FIN5	R/W	0h	Input filter selection for CELL Input 5 2 bits are used to define the edge filtering . 00 : No filtering 01 : Rising edge detect 10 : Falling edge detect 11 : Any edge detect Reset type: SYSRSn
9-8	FIN4	R/W	0h	Input filter selection for CELL Input 4 2 bits are used to define the edge filtering . 00 : No filtering 01 : Rising edge detect 10 : Falling edge detect 11 : Any edge detect Reset type: SYSRSn
7-6	FIN3	R/W	0h	Input filter selection for CELL Input 3 2 bits are used to define the edge filtering . 00 : No filtering 01 : Rising edge detect 10 : Falling edge detect 11 : Any edge detect Reset type: SYSRSn
5-4	FIN2	R/W	0h	Input filter selection for CELL Input 2 2 bits are used to define the edge filtering . 00 : No filtering 01 : Rising edge detect 10 : Falling edge detect 11 : Any edge detect Reset type: SYSRSn
3-2	FIN1	R/W	0h	Input filter selection for CELL Input 1 2 bits are used to define the edge filtering . 00 : No filtering 01 : Rising edge detect 10 : Falling edge detect 11 : Any edge detect Reset type: SYSRSn

Table 11-69. CLB_INPUT_FILTER Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	FIN0	R/W	0h	Input filter selection for CELL Input 0 2 bits are used to define the edge filtering . 00 : No filtering 01 : Rising edge detect 10 : Falling edge detect 11 : Any edge detect Reset type: SYSRSn

11.10.3.5 CLB_IN_MUX_SEL_0 Register (Offset = 10h) [Reset = 00000000h]

CLB_IN_MUX_SEL_0 is shown in Figure 11-65 and described in Table 11-70.

Return to the Summary Table.

Input selection to decide between Signals and GP register

Figure 11-65. CLB_IN_MUX_SEL_0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
SEL_GP_IN_7	SEL_GP_IN_6	SEL_GP_IN_5	SEL_GP_IN_4	SEL_GP_IN_3	SEL_GP_IN_2	SEL_GP_IN_1	SEL_GP_IN_0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 11-70. CLB_IN_MUX_SEL_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	SEL_GP_IN_7	R/W	0h	Select control for Input 7 to decide between external input and CLB_GP_REG[7] 0 : Input comes from selected external input 1 : Input comes from CLB_GP_REG[7] Reset type: SYSRSn
6	SEL_GP_IN_6	R/W	0h	Select control for Input 6 to decide between external input and CLB_GP_REG[6] 0 : Input comes from selected external input 1 : Input comes from CLB_GP_REG[6] Reset type: SYSRSn
5	SEL_GP_IN_5	R/W	0h	Select control for Input 5 to decide between external input and CLB_GP_REG[5] 0 : Input comes from selected external input 1 : Input comes from CLB_GP_REG[5] Reset type: SYSRSn
4	SEL_GP_IN_4	R/W	0h	Select control for Input 4 to decide between external input and CLB_GP_REG[4] 0 : Input comes from selected external input 1 : Input comes from CLB_GP_REG[4] Reset type: SYSRSn
3	SEL_GP_IN_3	R/W	0h	Select control for Input 3 to decide between external input and CLB_GP_REG[3] 0 : Input comes from selected external input 1 : Input comes from CLB_GP_REG[3] Reset type: SYSRSn
2	SEL_GP_IN_2	R/W	0h	Select control for Input 2 to decide between external input and CLB_GP_REG[2] 0 : Input comes from selected external input 1 : Input comes from CLB_GP_REG[2] Reset type: SYSRSn

Table 11-70. CLB_IN_MUX_SEL_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	SEL_GP_IN_1	R/W	0h	Select control for Input 1 to decide between external input and CLB_GP_REG[1] 0 : Input comes from selected external input 1 : Input comes from CLB_GP_REG[1] Reset type: SYSRSn
0	SEL_GP_IN_0	R/W	0h	Select control for Input 0 to decide between external input and CLB_GP_REG[0] 0 : Input comes from selected external input 1 : Input comes from CLB_GP_REG[0] Reset type: SYSRSn

11.10.3.6 CLB_LCL_MUX_SEL_1 Register (Offset = 14h) [Reset = 00000000h]

CLB_LCL_MUX_SEL_1 is shown in Figure 11-66 and described in Table 11-71.

Return to the Summary Table.

Input Mux selection for local mux

Figure 11-66. CLB_LCL_MUX_SEL_1 Register

31	30	29	28	27	26	25	24
MISC_INPUT_SEL_3	MISC_INPUT_SEL_2	MISC_INPUT_SEL_1	MISC_INPUT_SEL_0	RESERVED			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0-0h			
23	22	21	20	19	18	17	16
RESERVED				LCL_MUX_SEL_IN_3			
R-0-0h				R/W-0h			
15	14	13	12	11	10	9	8
LCL_MUX_SEL_IN_3	LCL_MUX_SEL_IN_2					LCL_MUX_SEL_IN_1	
R/W-0h	R/W-0h					R/W-0h	
7	6	5	4	3	2	1	0
LCL_MUX_SEL_IN_1			LCL_MUX_SEL_IN_0				
R/W-0h			R/W-0h				

Table 11-71. CLB_LCL_MUX_SEL_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MISC_INPUT_SEL_3	R/W	0h	When this bit is set to 1, the corresponding LCL_MUX_SEL_IN has a range of 32 to 63 Reset type: SYSRSn
30	MISC_INPUT_SEL_2	R/W	0h	When this bit is set to 1, the corresponding LCL_MUX_SEL_IN has a range of 32 to 63 Reset type: SYSRSn
29	MISC_INPUT_SEL_1	R/W	0h	When this bit is set to 1, the corresponding LCL_MUX_SEL_IN has a range of 32 to 63 Reset type: SYSRSn
28	MISC_INPUT_SEL_0	R/W	0h	When this bit is set to 1, the corresponding LCL_MUX_SEL_IN has a range of 32 to 63 Reset type: SYSRSn
27-20	RESERVED	R-0	0h	Reserved
19-15	LCL_MUX_SEL_IN_3	R/W	0h	5 bit MUX Select for Local MUX control for Input 3 See Local Signals and Mux Selection Table Reset type: SYSRSn
14-10	LCL_MUX_SEL_IN_2	R/W	0h	5 bit MUX Select for Local MUX control for Input 2 See Local Signals and Mux Selection Table Reset type: SYSRSn
9-5	LCL_MUX_SEL_IN_1	R/W	0h	5 bit MUX Select for Local MUX control for Input 1 See Local Signals and Mux Selection Table Reset type: SYSRSn
4-0	LCL_MUX_SEL_IN_0	R/W	0h	5 bit MUX Select for Local MUX control for Input 0 See Local Signals and Mux Selection Table Reset type: SYSRSn

11.10.3.7 CLB_LCL_MUX_SEL_2 Register (Offset = 18h) [Reset = 00000000h]

CLB_LCL_MUX_SEL_2 is shown in Figure 11-67 and described in Table 11-72.

Return to the Summary Table.

Input Mux selection for local mux

Figure 11-67. CLB_LCL_MUX_SEL_2 Register

31	30	29	28	27	26	25	24
MISC_INPUT_SEL_7	MISC_INPUT_SEL_6	MISC_INPUT_SEL_5	MISC_INPUT_SEL_4	RESERVED			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0-0h			
23	22	21	20	19	18	17	16
RESERVED				LCL_MUX_SEL_IN_7			
R-0-0h				R/W-0h			
15	14	13	12	11	10	9	8
LCL_MUX_SEL_IN_7	LCL_MUX_SEL_IN_6					LCL_MUX_SEL_IN_5	
R/W-0h	R/W-0h					R/W-0h	
7	6	5	4	3	2	1	0
LCL_MUX_SEL_IN_5			LCL_MUX_SEL_IN_4				
R/W-0h			R/W-0h				

Table 11-72. CLB_LCL_MUX_SEL_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MISC_INPUT_SEL_7	R/W	0h	When this bit is set to 1, the corresponding LCL_MUX_SEL_IN has a range of 32 to 63 Reset type: SYSRSn
30	MISC_INPUT_SEL_6	R/W	0h	When this bit is set to 1, the corresponding LCL_MUX_SEL_IN has a range of 32 to 63 Reset type: SYSRSn
29	MISC_INPUT_SEL_5	R/W	0h	When this bit is set to 1, the corresponding LCL_MUX_SEL_IN has a range of 32 to 63 Reset type: SYSRSn
28	MISC_INPUT_SEL_4	R/W	0h	When this bit is set to 1, the corresponding LCL_MUX_SEL_IN has a range of 32 to 63 Reset type: SYSRSn
27-20	RESERVED	R-0	0h	Reserved
19-15	LCL_MUX_SEL_IN_7	R/W	0h	5 bit MUX Select for Local MUX control for Input 7 See Local Signals and Mux Selection Table Reset type: SYSRSn
14-10	LCL_MUX_SEL_IN_6	R/W	0h	5 bit MUX Select for Local MUX control for Input 6 See Local Signals and Mux Selection Table Reset type: SYSRSn
9-5	LCL_MUX_SEL_IN_5	R/W	0h	5 bit MUX Select for Local MUX control for Input 5 See Local Signals and Mux Selection Table Reset type: SYSRSn
4-0	LCL_MUX_SEL_IN_4	R/W	0h	5 bit MUX Select for Local MUX control for Input 4 See Local Signals and Mux Selection Table Reset type: SYSRSn

11.10.3.8 CLB_BUF_PTR Register (Offset = 1Ch) [Reset = 00000000h]

CLB_BUF_PTR is shown in Figure 11-68 and described in Table 11-73.

Return to the Summary Table.

PUSH and PULL pointers

Figure 11-68. CLB_BUF_PTR Register

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

Table 11-73. CLB_BUF_PTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	Reserved
23-16	PUSH	R/W	0h	8 bit pointer which indicates the number of data values which have been pulled from the buffer by the High Level Controller. This counter will wrap around after 0xff. The Least significant 2 bits are used as the actual pointer for the operation. Reset type: SYSRSn
15-8	RESERVED	R-0	0h	Reserved
7-0	PULL	R/W	0h	8 bit pointer which indicates the number of data values that have been written by the High Level controller into the buffer. The Least significant 2 bits are used as the actual pointer for the operation. Reset type: SYSRSn

11.10.3.9 CLB_GP_REG Register (Offset = 20h) [Reset = 00000000h]

CLB_GP_REG is shown in Figure 11-69 and described in Table 11-74.

Return to the Summary Table.

General purpose register for CELL inputs

Figure 11-69. CLB_GP_REG Register

31	30	29	28	27	26	25	24
SW_RLS_CTRL_7	SW_RLS_CTRL_6	SW_RLS_CTRL_5	SW_RLS_CTRL_4	SW_RLS_CTRL_3	SW_RLS_CTRL_2	SW_RLS_CTRL_1	SW_RLS_CTRL_0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
SW_GATING_CTRL_7	SW_GATING_CTRL_6	SW_GATING_CTRL_5	SW_GATING_CTRL_4	SW_GATING_CTRL_3	SW_GATING_CTRL_2	SW_GATING_CTRL_1	SW_GATING_CTRL_0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
REG							
R/W-0h							

Table 11-74. CLB_GP_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SW_RLS_CTRL_7	R/W	0h	Software release control for output 7 of the asynchronous output conditioning block Reset type: SYSRSn
30	SW_RLS_CTRL_6	R/W	0h	Software release control for output 6 of the asynchronous output conditioning block Reset type: SYSRSn
29	SW_RLS_CTRL_5	R/W	0h	Software release control for output 5 of the asynchronous output conditioning block Reset type: SYSRSn
28	SW_RLS_CTRL_4	R/W	0h	Software release control for output 4 of the asynchronous output conditioning block Reset type: SYSRSn
27	SW_RLS_CTRL_3	R/W	0h	Software release control for output 3 of the asynchronous output conditioning block Reset type: SYSRSn
26	SW_RLS_CTRL_2	R/W	0h	Software release control for output 2 of the asynchronous output conditioning block Reset type: SYSRSn
25	SW_RLS_CTRL_1	R/W	0h	Software release control for output 1 of the asynchronous output conditioning block Reset type: SYSRSn
24	SW_RLS_CTRL_0	R/W	0h	Software release control for output 0 of the asynchronous output conditioning block Reset type: SYSRSn
23	SW_GATING_CTRL_7	R/W	0h	Software gating control for output 7 of the asynchronous output conditioning block Reset type: SYSRSn

Table 11-74. CLB_GP_REG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	SW_GATING_CTRL_6	R/W	0h	Software gating control for output 6 of the asynchronous output conditioning block Reset type: SYSRSn
21	SW_GATING_CTRL_5	R/W	0h	Software gating control for output 5 of the asynchronous output conditioning block Reset type: SYSRSn
20	SW_GATING_CTRL_4	R/W	0h	Software gating control for output 4 of the asynchronous output conditioning block Reset type: SYSRSn
19	SW_GATING_CTRL_3	R/W	0h	Software gating control for output 3 of the asynchronous output conditioning block Reset type: SYSRSn
18	SW_GATING_CTRL_2	R/W	0h	Software gating control for output 2 of the asynchronous output conditioning block Reset type: SYSRSn
17	SW_GATING_CTRL_1	R/W	0h	Software gating control for output 1 of the asynchronous output conditioning block Reset type: SYSRSn
16	SW_GATING_CTRL_0	R/W	0h	Software gating control for output 0 of the asynchronous output conditioning block Reset type: SYSRSn
15-8	RESERVED	R-0	0h	Reserved
7-0	REG	R/W	0h	8 bits which are directly connected to the 8 inputs of the CELL if that corresponding bit is selected in the CLB_IN_MUX_SEL_0 register Reset type: SYSRSn

11.10.3.10 CLB_OUT_EN Register (Offset = 24h) [Reset = 00000000h]

CLB_OUT_EN is shown in Figure 11-70 and described in Table 11-75.

Return to the Summary Table.

CELL output enable register

Figure 11-70. CLB_OUT_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUT0																															
R/W-0h																															

Table 11-75. CLB_OUT_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	OUT0	R/W	0h	32 bits which are directly driven out as OUTPUT_EN signals. Enabling bit x (x = 0:31) will override the corresponding peripheral signal muxed on the CLB OUTx. Reset type: SYSRSn

11.10.3.11 CLB_GLBL_MUX_SEL_1 Register (Offset = 28h) [Reset = 00000000h]

CLB_GLBL_MUX_SEL_1 is shown in Figure 11-71 and described in Table 11-76.

Return to the Summary Table.

Global Mux select for CELL inputs

Figure 11-71. CLB_GLBL_MUX_SEL_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				GLBL_MUX_SEL_IN_3							GLBL_MUX_SEL_IN_2				
R-0-0h				R/W-0h							R/W-0h				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GLBL_MUX_SEL_IN_2				GLBL_MUX_SEL_IN_1							GLBL_MUX_SEL_IN_0				
R/W-0h				R/W-0h							R/W-0h				

Table 11-76. CLB_GLBL_MUX_SEL_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R-0	0h	Reserved
27-21	GLBL_MUX_SEL_IN_3	R/W	0h	7 bit MUX Select for Global MUX control for Input 3 See Global Signals and Mux Selection Table Reset type: SYSRSn
20-14	GLBL_MUX_SEL_IN_2	R/W	0h	7 bit MUX Select for Global MUX control for Input 2 See Global Signals and Mux Selection Table Reset type: SYSRSn
13-7	GLBL_MUX_SEL_IN_1	R/W	0h	7 bit MUX Select for Global MUX control for Input 1 See Global Signals and Mux Selection Table Reset type: SYSRSn
6-0	GLBL_MUX_SEL_IN_0	R/W	0h	7 bit MUX Select for Global MUX control for Input 0 See Global Signals and Mux Selection Table Reset type: SYSRSn

11.10.3.12 CLB_GLBL_MUX_SEL_2 Register (Offset = 2Ch) [Reset = 00000000h]

CLB_GLBL_MUX_SEL_2 is shown in Figure 11-72 and described in Table 11-77.

Return to the Summary Table.

Global Mux select for CELL inputs

Figure 11-72. CLB_GLBL_MUX_SEL_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				GLBL_MUX_SEL_IN_7							GLBL_MUX_SEL_IN_6				
R-0-0h				R/W-0h							R/W-0h				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GLBL_MUX_SEL_IN_6				GLBL_MUX_SEL_IN_5							GLBL_MUX_SEL_IN_4				
R/W-0h				R/W-0h							R/W-0h				

Table 11-77. CLB_GLBL_MUX_SEL_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R-0	0h	Reserved
27-21	GLBL_MUX_SEL_IN_7	R/W	0h	7 bit MUX Select for Global MUX control for Input 7 See Global Signals and Mux Selection Table Reset type: SYSRSn
20-14	GLBL_MUX_SEL_IN_6	R/W	0h	7 bit MUX Select for Global MUX control for Input 6 See Global Signals and Mux Selection Table Reset type: SYSRSn
13-7	GLBL_MUX_SEL_IN_5	R/W	0h	7 bit MUX Select for Global MUX control for Input 5 See Global Signals and Mux Selection Table Reset type: SYSRSn
6-0	GLBL_MUX_SEL_IN_4	R/W	0h	7 bit MUX Select for Global MUX control for Input 4 See Global Signals and Mux Selection Table Reset type: SYSRSn

11.10.3.13 CLB_PRESCALE_CTRL Register (Offset = 30h) [Reset = 00000000h]

CLB_PRESCALE_CTRL is shown in Figure 11-73 and described in Table 11-78.

Return to the Summary Table.

Prescaler register control

Figure 11-73. CLB_PRESCALE_CTRL Register

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

Table 11-78. CLB_PRESCALE_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PRESCALE	R/W	0h	16-bit Value of prescaler to be used for the counter as reference to reset when reaching this value. The counter is a simple incrementing counter which will count up to the reference value and reset to 0 and this cycle will continue as long as the counter is enabled. This 16-bit register value is used as a reference for the 16-bit counter to reset to zero whenever count reaches this value. Reset type: SYSRSn
15-6	RESERVED	R-0	0h	Reserved
5-2	TAP	R/W	0h	TAP Select value. These 4 bits will be used as a select to tap one of the 16 register bit position of the counter as the output. 0000 selects Counter Bit position 0 0001 selects Counter Bit position 1 .... 1111 selects Counter Bit position 15 Reset type: SYSRSn
1	STRB	R/W	0h	When set to 0, a strobe output will be sent out whenever the counter value matches the PRESCALE_VALUE. When set to 1, the output of the counter register bit position as selected by TAP_SELECT_VALUE will be sent out. Reset type: SYSRSn
0	CLKEN	R/W	0h	Enable the prescale clock/strobe generator. A 16-bit counter is used to either generate a strobe or send out a selected counter bit position to the CLB CELL. This is meant to be a general purpose strobe/prescaled clock which can be used by the CELL logic if needed. This will be sent to the CELL through one of the LCL_IN MUX ports. Reset type: SYSRSn

11.10.3.14 CLB_INTR_TAG_REG Register (Offset = 40h) [Reset = 0000h]

CLB_INTR_TAG_REG is shown in Figure 11-74 and described in Table 11-79.

Return to the Summary Table.

Interrupt Tag register

Figure 11-74. CLB_INTR_TAG_REG Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		TAG					
R-0-0h		R/W-0h					

Table 11-79. CLB_INTR_TAG_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-6	RESERVED	R-0	0h	Reserved
5-0	TAG	R/W	0h	6 bits which are used by the High Level Controller to set a tag value on flagging interrupts. This can be cleared through the VBUS interface since it is writeable through the VBUS. Reset type: SYSRSn

11.10.3.15 CLB_LOCK Register (Offset = 44h) [Reset = 00000000h]

CLB_LOCK is shown in Figure 11-75 and described in Table 11-80.

Return to the Summary Table.

Lock control register

Figure 11-75. CLB_LOCK Register

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

Table 11-80. CLB_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	WSonce	0h	These 16 bits act as a key to enable writes to Bit 0 of this register. The only time a '1' can be written to Bit 0 is by a single 32-bit write where bits 31:16 equal 0x5a5a and bit 0 is '1'. All other writes are ignored including separate 16-bit writes. This is EALLOW protected. Reset type: SYSRSn
15-1	RESERVED	R-0	0h	Reserved
0	LOCK	R/W	0h	This bit is used as a one-time write bit (Set Once). Once it is set to '1', only a reset (SYSRSN 0) will clear this bit back to 0. Reset type: SYSRSn

11.10.3.16 CLB_HLC_INSTR_READ_PTR Register (Offset = 48h) [Reset = 0000h]

CLB_HLC_INSTR_READ_PTR is shown in Figure 11-76 and described in Table 11-81.

Return to the Summary Table.

HLC instruction read pointer

Figure 11-76. CLB_HLC_INSTR_READ_PTR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED			READ_PTR				
R-0-0h			R/W-0h				

Table 11-81. CLB_HLC_INSTR_READ_PTR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R-0	0h	Reserved
4-0	READ_PTR	R/W	0h	This is a 5 bit value which will be used as an address pointer to read out HLC instruction memory. Reset type: SYSRSn

11.10.3.17 CLB_HLC_INSTR_VALUE Register (Offset = 4Ch) [Reset = 0000h]

CLB_HLC_INSTR_VALUE is shown in Figure 11-77 and described in Table 11-82.

Return to the Summary Table.

HLC instruction read value

Figure 11-77. CLB_HLC_INSTR_VALUE Register

15	14	13	12	11	10	9	8
RESERVED				INSTR			
R-0-0h				R-0h			
7	6	5	4	3	2	1	0
INSTR							
R-0h							

Table 11-82. CLB_HLC_INSTR_VALUE Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R-0	0h	Reserved
11-0	INSTR	R	0h	This is a 12 bit value which will read the content of the HLC instruction memory address pointed by CLB_HLC_INSTR_READ_PTR register. Reset type: SYSRSn

11.10.3.18 CLB_DBG_OUT_2 Register (Offset = 5Ch) [Reset = 00000000h]

CLB_DBG_OUT_2 is shown in Figure 11-78 and described in Table 11-83.

Return to the Summary Table.

Visibility for CLB inputs and final asynchronous outputs

Figure 11-78. CLB_DBG_OUT_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IN								OUT							
R/W1C-0h																R/W-0h								R/W-0h							

Table 11-83. CLB_DBG_OUT_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W1C	0h	Reserved
15-8	IN	R/W	0h	These bits reflect the state of the 8 inputs finally going to the CELL after selection and input conditioning. Reset type: SYSRSn
7-0	OUT	R/W	0h	These bits reflect the state of the 8 outputs of the Output Conditioning Block. Reset type: SYSRSn

11.10.3.19 CLB_DBG_R0 Register (Offset = 60h) [Reset = 00000000h]

CLB_DBG_R0 is shown in Figure 11-79 and described in Table 11-84.

Return to the Summary Table.

R0 of High level Controller

Figure 11-79. CLB_DBG_R0 Register

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

Table 11-84. CLB_DBG_R0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DBG	R	0h	CLB_DBG_R0 Reset type: SYSRSn

11.10.3.20 CLB_DBG_R1 Register (Offset = 64h) [Reset = 00000000h]

CLB_DBG_R1 is shown in Figure 11-80 and described in Table 11-85.

Return to the Summary Table.

R1 of High level Controller

Figure 11-80. CLB_DBG_R1 Register

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

Table 11-85. CLB_DBG_R1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DBG	R	0h	CLB_DBG_R1 Reset type: SYSRSn

11.10.3.21 CLB_DBG_R2 Register (Offset = 68h) [Reset = 00000000h]

CLB_DBG_R2 is shown in Figure 11-81 and described in Table 11-86.

Return to the Summary Table.

R2 of High level Controller

Figure 11-81. CLB_DBG_R2 Register

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

Table 11-86. CLB_DBG_R2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DBG	R	0h	CLB_DBG_R2 Reset type: SYSRSn

11.10.3.22 CLB_DBG_R3 Register (Offset = 6Ch) [Reset = 00000000h]

CLB_DBG_R3 is shown in Figure 11-82 and described in Table 11-87.

Return to the Summary Table.

R3 of High level Controller

Figure 11-82. CLB_DBG_R3 Register

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

Table 11-87. CLB_DBG_R3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DBG	R	0h	CLB_DBG_R3 Reset type: SYSRSn

11.10.3.23 CLB_DBG_C0 Register (Offset = 70h) [Reset = 00000000h]

CLB_DBG_C0 is shown in Figure 11-83 and described in Table 11-88.

Return to the Summary Table.

Count of Unit 0

Figure 11-83. CLB_DBG_C0 Register

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

Table 11-88. CLB_DBG_C0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DBG	R	0h	CLB_DBG_C0 Reset type: SYSRSn

11.10.3.24 CLB_DBG_C1 Register (Offset = 74h) [Reset = 00000000h]

CLB_DBG_C1 is shown in Figure 11-84 and described in Table 11-89.

Return to the Summary Table.

Count of Unit 1

Figure 11-84. CLB_DBG_C1 Register

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

Table 11-89. CLB_DBG_C1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DBG	R	0h	CLB_DBG_C1 Reset type: SYSRSn

11.10.3.25 CLB_DBG_C2 Register (Offset = 78h) [Reset = 00000000h]

CLB_DBG_C2 is shown in Figure 11-85 and described in Table 11-90.

Return to the Summary Table.

Count of Unit 2

Figure 11-85. CLB_DBG_C2 Register

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

Table 11-90. CLB_DBG_C2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DBG	R	0h	CLB_DBG_C2 Reset type: SYSRSn

11.10.3.26 CLB_DBG_OUT Register (Offset = 7Ch) [Reset = 00010100h]

CLB_DBG_OUT is shown in Figure 11-86 and described in Table 11-91.

Return to the Summary Table.

Outputs of various units in the Cell

Figure 11-86. CLB_DBG_OUT Register

31	30	29	28	27	26	25	24
OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
LUT42_OUT	FSM2_LUTOUT	FSM2_S1	FSM2_S0	COUNT2_MATCH1	COUNT2_ZERO	COUNT2_MATCH2	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-1-1h
15	14	13	12	11	10	9	8
LUT41_OUT	FSM1_LUTOUT	FSM1_S1	FSM1_S0	COUNT1_MATCH1	COUNT1_ZERO	COUNT1_MATCH2	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-1-1h
7	6	5	4	3	2	1	0
LUT40_OUT	FSM0_LUTOUT	FSM0_S1	FSM0_S0	COUNT0_MATCH1	COUNT0_ZERO	COUNT0_MATCH2	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0-0h

Table 11-91. CLB_DBG_OUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	OUT7	R	0h	CELL Output 7 Reset type: SYSRSn
30	OUT6	R	0h	CELL Output 6 Reset type: SYSRSn
29	OUT5	R	0h	CELL Output 5 Reset type: SYSRSn
28	OUT4	R	0h	CELL Output 4 Reset type: SYSRSn
27	OUT3	R	0h	CELL Output 3 Reset type: SYSRSn
26	OUT2	R	0h	CELL Output 2 Reset type: SYSRSn
25	OUT1	R	0h	CELL Output 1 Reset type: SYSRSn
24	OUT0	R	0h	CELL Output 0 Reset type: SYSRSn
23	LUT42_OUT	R	0h	LUT4_OUT UNIT 2 Reset type: SYSRSn
22	FSM2_LUTOUT	R	0h	FSM_LUT_OUT UNIT 2 Reset type: SYSRSn
21	FSM2_S1	R	0h	FSM_S1 UNIT 2 Reset type: SYSRSn
20	FSM2_S0	R	0h	FSM_S0 UNIT 2 Reset type: SYSRSn

Table 11-91. CLB_DBG_OUT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	COUNT2_MATCH1	R	0h	COUNT_MATCH1 UNIT 2 Reset type: SYSRSn
18	COUNT2_ZERO	R	0h	COUNT_ZERO UNIT 2 Reset type: SYSRSn
17	COUNT2_MATCH2	R	0h	COUNT_MATCH2 UNIT 2 Reset type: SYSRSn
16	RESERVED	R-1	1h	Reserved
15	LUT41_OUT	R	0h	LUT4_OUT UNIT 1 Reset type: SYSRSn
14	FSM1_LUTOUT	R	0h	FSM_LUT_OUT UNIT 1 Reset type: SYSRSn
13	FSM1_S1	R	0h	FSM_S1 UNIT 1 Reset type: SYSRSn
12	FSM1_S0	R	0h	FSM_S0 UNIT 1 Reset type: SYSRSn
11	COUNT1_MATCH1	R	0h	COUNT_MATCH1 UNIT 1 Reset type: SYSRSn
10	COUNT1_ZERO	R	0h	COUNT_ZERO UNIT 1 Reset type: SYSRSn
9	COUNT1_MATCH2	R	0h	COUNT_MATCH2 UNIT 1 Reset type: SYSRSn
8	RESERVED	R-1	1h	Reserved
7	LUT40_OUT	R	0h	LUT4_OUT UNIT 0 Reset type: SYSRSn
6	FSM0_LUTOUT	R	0h	FSM_LUT_OUT UNIT 0 Reset type: SYSRSn
5	FSM0_S1	R	0h	FSM_S1 UNIT 0 Reset type: SYSRSn
4	FSM0_S0	R	0h	FSM_S0 UNIT 0 Reset type: SYSRSn
3	COUNT0_MATCH1	R	0h	COUNT_MATCH1 UNIT 0 Reset type: SYSRSn
2	COUNT0_ZERO	R	0h	COUNT_ZERO UNIT 0 Reset type: SYSRSn
1	COUNT0_MATCH2	R	0h	COUNT_MATCH2 UNIT 0 Reset type: SYSRSn
0	RESERVED	R-0	0h	Reserved

11.10.4 CLB_DATA_EXCHANGE_REGS Registers

Table 11-92 lists the memory-mapped registers for the CLB_DATA_EXCHANGE_REGS registers. All register offset addresses not listed in Table 11-92 should be considered as reserved locations and the register contents should not be modified.

Table 11-92. CLB_DATA_EXCHANGE_REGS Registers

Offset	Acronym	Register Name	Protection
0h	CLB_PUSH	CLB_PUSH FIFO Registers (from HLC)	
80h	CLB_PULL	CLB_PULL FIFO Registers (TO HLC)	

Complex bit access types are encoded to fit into small table cells. Table 11-93 shows the codes that are used for access types in this section.

Table 11-93. CLB_DATA_EXCHANGE_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

11.10.4.1 CLB_PUSH Register (Offset = 0h) [Reset = 00000000h]

CLB_PUSH is shown in Figure 11-87 and described in Table 11-94.

Return to the Summary Table.

CLB_PUSH FIFO Registers (from HLC)

Figure 11-87. CLB_PUSH Register

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

Table 11-94. CLB_PUSH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PUSH	R	0h	FIFO TO System From CLB Reset type: SYSRSn

11.10.4.2 CLB_PULL Register (Offset = 80h) [Reset = 00000000h]

CLB_PULL is shown in Figure 11-88 and described in Table 11-95.

Return to the Summary Table.

CLB_PULL FIFO Registers (TO HLC)

Figure 11-88. CLB_PULL Register

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

Table 11-95. CLB_PULL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PULL	R/W	0h	FIFO From system TO CLB Reset type: SYSRSn

Chapter 12

Dual-Clock Comparator (DCC)

This chapter describes the Dual-Clock Comparator (DCC) module.

12.1 Introduction

The dual-clock comparator module is used for evaluating and monitoring the clock input based on a second clock, which can be a more accurate and reliable version. This instrumentation is used to detect faults in clock source or clock structures, thereby enhancing the system's safety metrics.

12.1.1 Features

The main features of each of the DCC modules are:

- Allows the application to make sure that a fixed ratio is maintained between frequencies of two clock signals.
- Supports the definition of a programmable tolerance window in terms of the number of reference clock cycles.
- Supports continuous monitoring without requiring application intervention.
- Supports a single-sequence mode for spot measurements.
- Allows the selection of a clock source for each of the counters, resulting in several specific use cases.

12.1.2 Block Diagram

Figure 12-1 shows how the DCC connects to the rest of the system. Figure 12-2 shows the main concept of the DCC module.

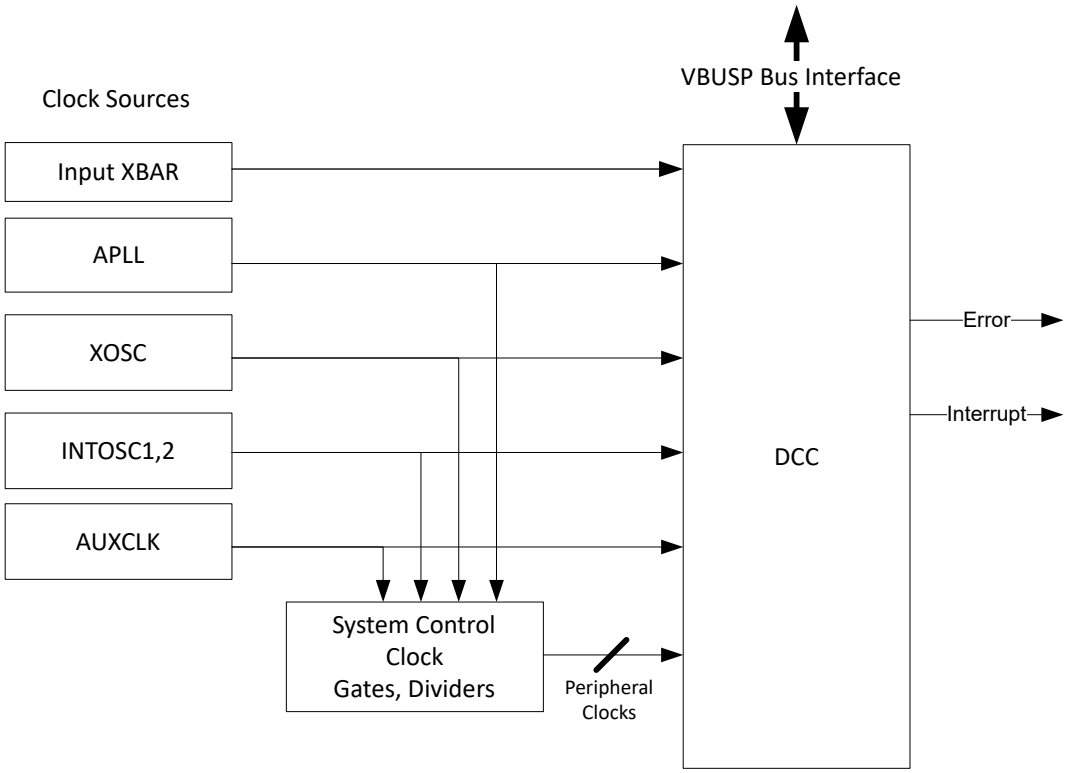


Figure 12-1. DCC Module Overview

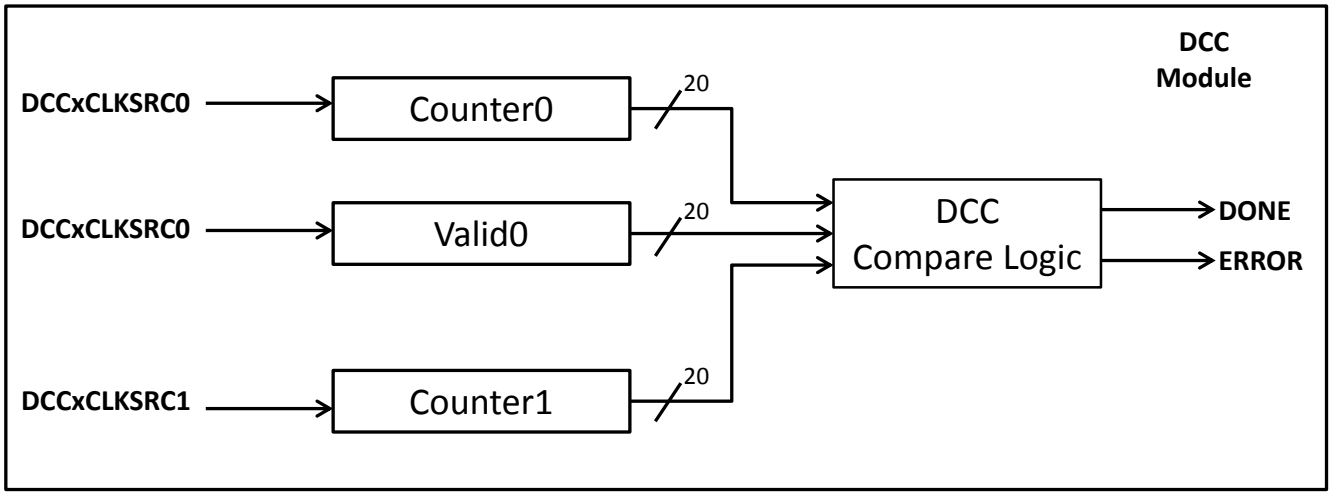


Figure 12-2. DCC Operation

12.2 Module Operation

As shown in Figure 12-2, DCC contains three counters – Counter0, Valid0 and Counter1. Initially, all counters are loaded with the user-defined, pre-load value. Counter0 and Counter1 start decrementing once the DCC is enabled at rates determined by the frequencies of Clock0 and Clock1, respectively. When Counter0 equals 0 (expires), the Valid0 counter decrements at a rate determined by Clock0. If Counter1 decrements to 0 in the valid window, then no error is generated and Clock1 is considered to be good within allowable tolerance as configured by the user.

12.2.1 Configuring DCC Counters

Counter0 and Counter1 are configured based on the ratio between the frequencies of Clock0 and Clock1 (Fclk1 × Counter0 = Fclk0×Counter1). The Valid0 counter provides tolerance and is configured based on the error in DCC. Since Clock0 and Clock1 are asynchronous, the start and stop of the counters do not occur synchronously. Hence, while configuring the counters, two different sources of errors must be accounted for:

- DCC Errors due to the asynchronous timing of Clock0 and Clock1: this depends on the frequency of Clock0 and Clock1:
  - If Fclk1 > Fclk0, then Async. Error (in Clock0 cycles) = 2 + 2 × (Fsysclk/Fclk0)
  - If Fclk1 < Fclk0, then Async. Error (in Clock0 cycles) = 2 × (Fclk0/Fclk1) + 2 × (Fsysclk/Fclk0)
  - If Fclk1 is unknown, then Async. Error (in Clock0 cycles) = 2 + 2 × (Fsysclk/Fclk0)
- Digitization Error = 8 Clock0 cycles

DCC Error (in Clock0 Cycles) = Async. Error + Digitization Error

DCC error shows up as a frequency error for clock under measurement. This error is DCC induced and does not represent error in frequency of clock under measurement. The application needs to take this into consideration while configuring the counters, and determine a desirable tolerance for DCC error that defines the window of measurement. To illustrate:

Window (in Clock0 Cycles) = (DCC Error)/(0.01 × Tolerance)

For example, if DCC Error is 10 and the tolerance desired is ±0.1%, then:

Window (in Clock0 Cycles) = 10/(0.01 × 0.1) = 10000

Based on above formula for Window, if the desired tolerance is low, then the counter values are large and increase the window of measurement. This means that counter values for a tolerance of 0.1% are larger than that of 0.2%. So, based on the application defined tolerance, define the window of measurement in terms of Clock0 cycles.

The clock under measurement can have an allowed frequency error. If this error is expected, then the error can also be accounted while configuring counters. For example, if measuring INTOSC1/2 frequency using an external crystal as a reference clock, the allowable tolerance of INTOSC1/2 (for example, ±1%) can be accounted for and factored into the counter configuration. The formula is:

Frequency Error Allowed (in Clock0 Cycles) = Window × (Allowable Frequency Tolerance (in %) / 100)

Total Error (in Clock0 Cycles) = DCC Error + Frequency Error Allowed

The following equations are used to configure counter values:

Counter0 (DCCCNTSEED0) = Window - Total Error

Valid0 (DCCVALIDSEED0) = 2 × Total Error

Counter1 (DCCCNTSEED1) = Window × (Fclk1/Fclk0)

Note

Counter1 is a 20-bit counter, so the maximum possible value cannot exceed 1048575. If the value does exceed, then increase the desired Tolerance for DCC error, so that Window of measurement is lowered. The following formula can be used to compute minimum tolerance possible:

Tolerance (%) = (100 × DCC Error × (Fclk1/Fclk0)) / 1048575

12.2.2 Single-Shot Measurement Mode

The DCC module can be programmed to count down one time by enabling the single-shot mode. In this mode, the DCC stops operating when the down counter0 and the valid counter0 reach 0.

At the end of one sequence of counting down in this single-shot mode, the DCC gets disabled automatically, which prevents further counting. This mode is typically used for spot-checking the frequency of a signal.

Example-1: Validating PLLRAWCLK frequency

A practical example of the usage is to validate the PLL output clock frequency using the XTAL as the reference clock. Assume XTAL is 10MHz, PLL output frequency is 100MHz, SYSCLK is 100MHz, allowable Frequency Tolerance is 0.1%, and DCC Tolerance required is 0.1%. The measurement sequence proceeds as follows:

- Set Clock0 source for Counter0 and Valid0 as XTAL, and Clock1 source for Counter1 as PLL output clock.
- Based on the equations defined in Section 12.2.1, calculated seed values for Counters can be Counter0 = 29940; Valid0 = 120; Counter1 = 300000
- Once the DCC is enabled, the counters Counter0 and Counter1 both start counting down from the seed values.
- When Counter0 reaches zero, Counter0 automatically triggers the Valid0 counter.
- When Valid0 reaches zero and Counter1 is not zero, an ERROR status flag is set and a "DCC error" is sent to the Interrupt Controller. Counter1 is frozen so that the counter stops counting down any further. The application can enable an interrupt to be generated from the Interrupt Controller whenever this DCC error is indicated.
- The application then needs to clear the ERROR status flag and restart the DCC module so that the module is ready for the next spot measurement.

If there is no error generated at the end of the sequence, then the DONE status flag is set and a DONE interrupt is generated. The application must clear the DONE flag before restarting the DCC.

Error Conditions:

An error condition is generated by any one of the following:

1. Counter1 counts down to 0 before Counter0 reaches 0. This means that Clock1 is faster than expected, or Clock0 is slower than expected. This error includes the case when Clock0 is stuck at 1 or 0.
2. Counter1 does not reach 0 even when Counter0 and Valid0 have both reached 0. This means that Clock1 is slower than expected. This error includes the case when Clock1 is stuck at 1 or 0.

Any error freezes the counters from counting. An application can then read out the counter values to help determine what caused the error.

Example-2: Measuring AUXCLKIN frequency

Another example of single-shot mode is to measure the frequency of AUXCLKIN (unknown frequency) using INTOSC1 (10MHz) as the reference clock and SYSCLK is 10MHz. The measurement sequence proceeds as follows:

- Set Clock0 source for Counter0 and Valid0 as INTOSC1 (10MHz), and Clock1 source for Counter1 as AUXCLKIN.
- Now configure counter values using equations in Section 12.2.1. For tolerance = ±0.1%, Total Error = 10 clock0 cycles; Window = 10000 clock0 cycles; Counter0 = 9990; Valid0 = 20. Since Clock1 frequency (Fclk1) is unknown, the Counter1 value can be set to the maximum value, 1048575 (0xFFFFF).
- Once the DCC is enabled, the counters Counter0 and Counter1 both start counting down from the seed values.
- Since Counter1 is set to the maximum value, 1048575, the counter does not expire when Counter0 and Valid0 have expired. This generates an error that is expected and the application ignores this error and uses Counter1 values to compute the frequency of Clock1 (Fclk1).
- Knowing the frequency of Clock0 (INTOSC1), Fclk0 = 10MHz, and using Equation 1, the frequency of AUXCLKIN, Fclk1, can be measured:

$$Fclk1 = \frac{Fclk0 \times (1048575 - Meas.Counter1)}{(Counter0 + Valid0)} = \frac{10 \times (1048575 - Meas.Counter1)}{(9990 + 20)} \quad (1)$$

12.2.3 Continuous Monitoring Mode

In this mode, the DCC is used by the application to make sure that two clock signals maintain the correct frequency ratio. Suppose the application wants to make sure that the PLL output signal always maintains a fixed frequency relationship with the XTAL:

- In this case, the application can use the XTAL as the Clock0 signal (for Counter0 and Valid0) and the PLL output as the Clock1 (for Counter1).
- The seed values of Counter0, Valid0 and Counter1 are selected based on the equations defined in Section 12.2.1 such that if the actual frequencies of Clock0 and Clock1 are equal to the expected frequencies, then the Counter1 reaches zero during the count down of the Valid0 counter.
- If the Counter1 reaches zero during the count down of the Valid0 counter, then all the counters (Counter0, Valid0, Counter1) are reloaded with the initial seed values.
- This sequence of counting down and checking then continues as long as there is no error, or until the DCC module is disabled.
- The counters must get reloaded if the application resets and restarts the DCC module.

Error Conditions:

An error condition is generated by one of the following:

1. Counter1 counts down to 0 before Counter0 reaches 0. This means that Clock1 is faster than expected or Clock0 is slower than expected. This condition includes the case when Clock0 is stuck at 1 or 0.
2. Counter1 does not reach 0 even when Counter0 and Valid0 have both reached 0. This means that Clock1 is slower than expected. This condition includes the case when Clock1 is stuck at 1 or 0.

Any error freezes the counters from counting. An application can then read out the counter values to help determine what caused the error.

12.2.4 Error Conditions

While operating in continuous mode, the counters get reloaded with the seed values and continue counting down under the following conditions:

- The module is reset or restarted by the application, OR
- Counter0, Valid 0, and Counter1 all reach 0 without any error.

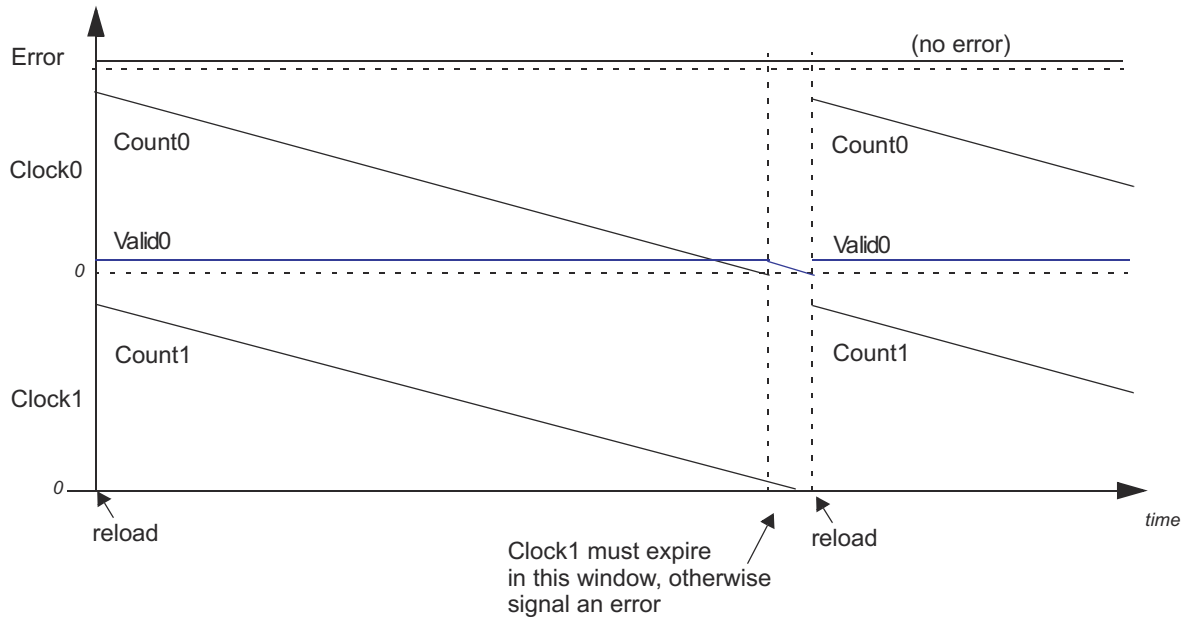


Figure 12-3. Counter Relationship

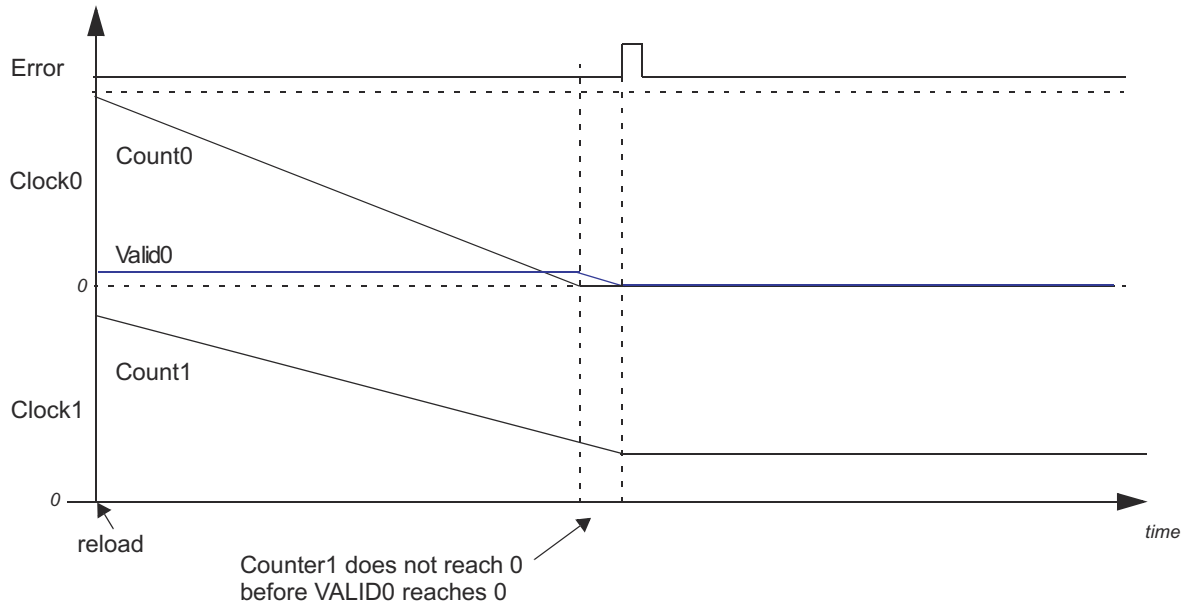


Figure 12-4. Clock1 Slower Than Clock0 - Results in an Error and Stops Counting

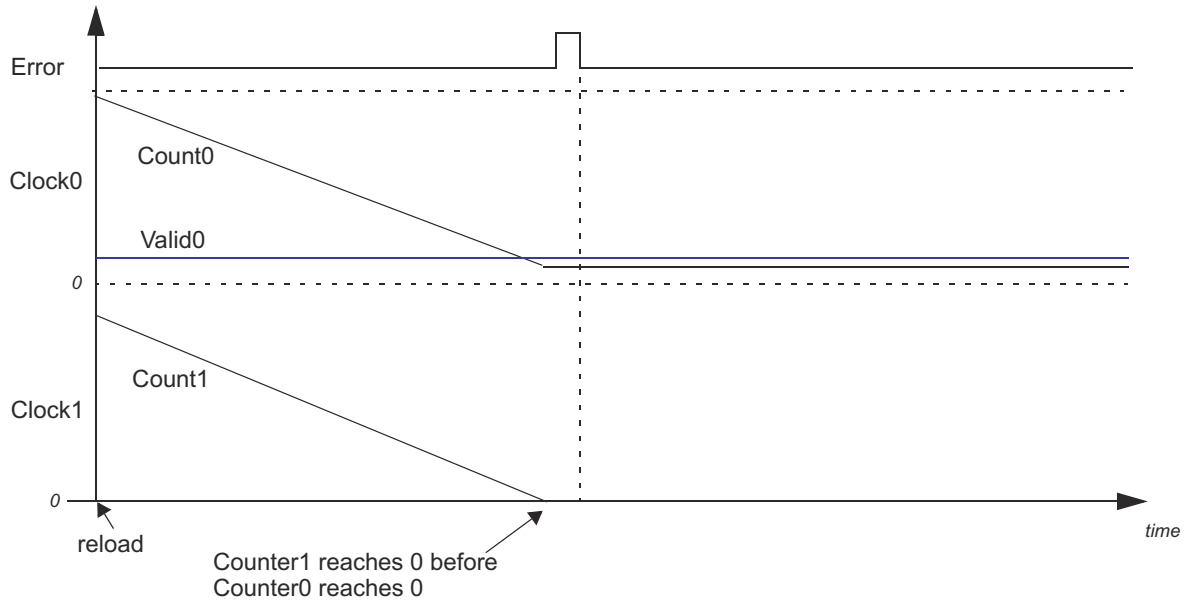


Figure 12-5. Clock1 Faster Than Clock0 - Results in an Error and Stops Counting

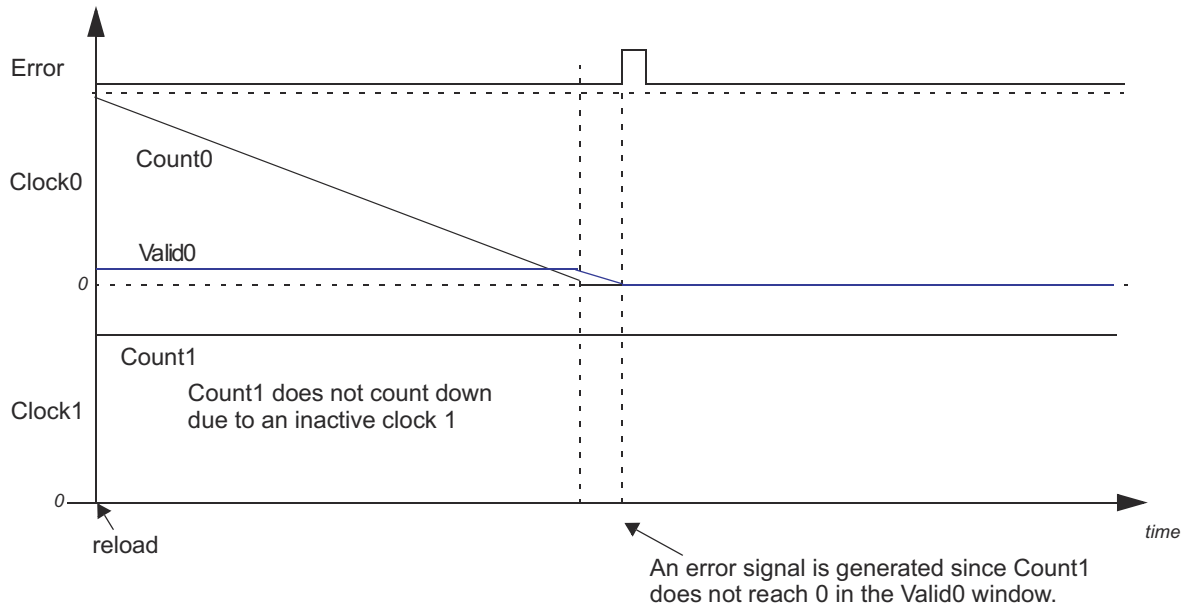


Figure 12-6. Clock1 Not Present - Results in an Error and Stops Counting

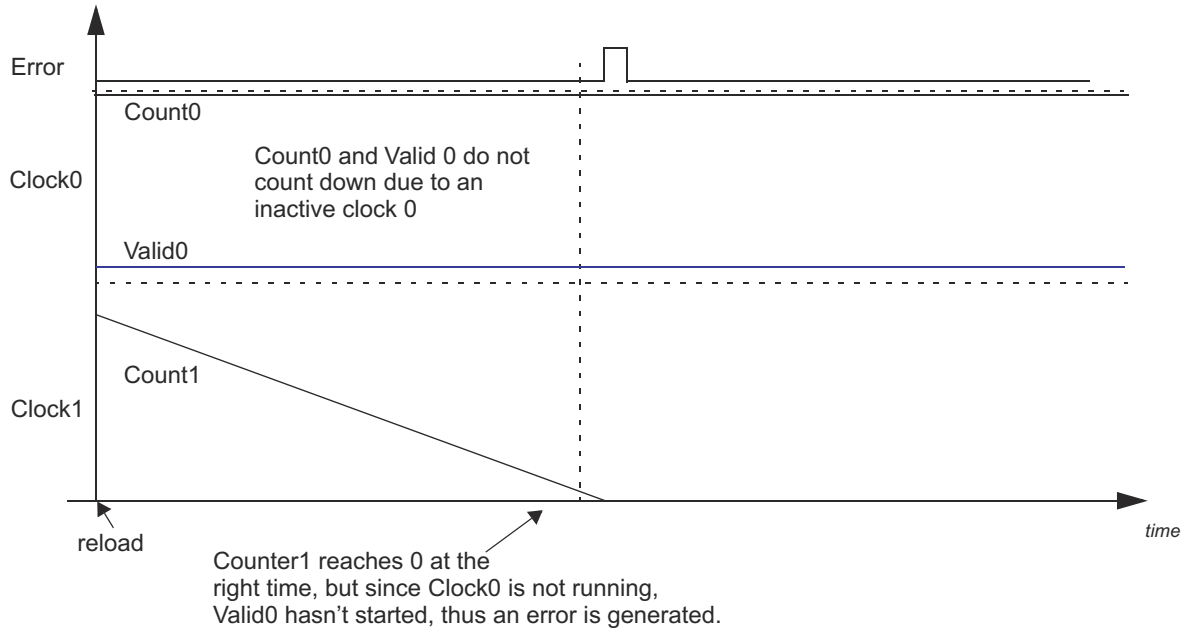


Figure 12-7. Clock0 Not Present - Results in an Error and Stops Counting

12.3 Interrupts

DCC generates an interrupt on either of two events:

- DCC finishes counting and all the counters expire within a defined window indicating DONE operation, provided DCCGCTRL.DONENA = 1.
- DCC finishes counting with error where counters do not expire in a defined window. This indicates an ERROR event, and sets an interrupt provided DCCGCTRL.ERRENA = 1.

Interrupts generated by DONE or ERROR events are ORed and flagged as a DCCn_DONE or DCCn_ERROR interrupt, respectively. Refer to PIPE Channel Mapping table in the *System Control and Interrupts* chapter determine the interrupt channel mapping. The application interrupt service routine needs to check the status flag inside the DCCSTATUS register to determine whether the interrupt is due to ERROR or DONE.

12.4 Software

12.4.1 DCC Registers to Driverlib Functions

Table 12-1. DCC Registers to Driverlib Functions

File	Driverlib Function
DCCGCTRL	
dcc.h	DCC_enableModule
dcc.h	DCC_disableModule
dcc.h	DCC_enableErrorSignal
dcc.h	DCC_enableDoneSignal
dcc.h	DCC_disableErrorSignal
dcc.h	DCC_disableDoneSignal
dcc.h	DCC_enableSingleShotMode
dcc.h	DCC_disableSingleShotMode
DCCCNTSEED0	
dcc.h	DCC_setCounterSeeds
DCCVALIDSEED0	
dcc.h	DCC_setCounterSeeds
DCCCNTSEED1	
dcc.h	DCC_setCounterSeeds
DCCSTATUS	
dcc.h	DCC_getErrorStatus
dcc.h	DCC_getSingleShotStatus
dcc.h	DCC_clearErrorFlag
dcc.h	DCC_clearDoneFlag
sysctl.c	SysCtl_isPLLValid
DCCCNT0	
dcc.h	DCC_getCounter0Value
DCCVALID0	
dcc.h	DCC_getValidCounter0Value
DCCCNT1	
dcc.h	DCC_getCounter1Value
DCCCLKSRC1	
dcc.h	DCC_setCounter1ClkSource
dcc.h	DCC_getCounter1ClkSource
DCCCLKSRC0	
dcc.h	DCC_setCounter0ClkSource
dcc.h	DCC_getCounter0ClkSource

12.4.2 DCC Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/dcc

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

12.4.2.1 DCC Single shot Clock verification - SINGLE_CORE

FILE: dcc_ex1_single_shot_verification.c

This program uses the XTAL oscillator as a reference clock to verify the frequency of the PLL.

The Dual-Clock Comparator Module 1 is used for the clock verification. The clocksource0 is the reference clock (Fclk0 = 25Mhz) and the clocksource1 is the clock that needs to be verified (Fclk1 = 200Mhz). Seed is the value that gets loaded into the Counter.

Please refer to the TRM for details on counter seed values to be set.

External Connections

- None

Watch Variables

- *result* - Status of the clock verification

12.4.2.2 DCC Single shot Clock measurement - SINGLE_CORE

FILE: dcc_ex2_single_shot_measurement.c

This program demonstrates Single Shot measurement of the INTOSC1 clock post trim using XTAL as the reference clock.

The Dual-Clock Comparator Module 1 is used for the clock measurement. The clocksource0 is the reference clock (Fclk0 = 25Mhz) and the clocksource1 is the clock that needs to be measured (Fclk1 = 10Mhz). Since the frequency of clock1 needs to be measured, an initial seed is set to the max value of the counter.

Please refer to the TRM for details on counter seed values to be set.

External Connections

- None

Watch Variables

- *result* - Status if the INTOSC1 clock measurement completed successfully.
- *meas_freq1* - measured clock frequency, in this case for INTOSC1.

12.4.2.3 DCC Continuous clock monitoring - SINGLE_CORE

FILE: dcc_ex3_continuous_monitoring_of_clock.c

This program demonstrates continuous monitoring of PLL Clock in the system using INTOSC1 as the reference clock. This would trigger an error signal on any error, causing the decrement/ reload of counters to stop. The Dual-Clock Comparator Module 1 is used for the clock monitoring. The clocksource0 is the reference clock (Fclk0 = 10Mhz) and the clocksource1 is the clock that needs to be monitored (Fclk1 = 200Mhz). The clock0 and clock1 seed are set automatically by the error tolerances defined in the sysconfig file included in this project. For the sake of demo an un-realistic tolerance is assumed to generate an error on continuous monitoring.

Please refer to the TRM for details on counter seed values to be set. Note : When running in flash configuration it is good to do a reset & restart after loading the example to remove any stale flags/states.

External Connections

- None

Watch Variables

- *result* - Status of the PLLRAW clock monitoring
- *cnt0* - Counter0 Value measure when error is generated
- *cnt1* - Counter1 Value measure when error is generated
- *valid* - Valid0 Value measure when error is generated
- *esm0IntCount* - Indication that ESM Low Priority Int ISR is entered
- *highestIntr* - ESM Global event number of the highest priority outstanding low priority interrupt that is pending - corresponds to DCCx_ERR for this example

12.5 DCC Registers

This section describes the DCC Registers.

12.5.1 DCC Base Address Table

Table 12-2. DCC Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
DCC_REGS	DCC1_BASE	0x6008_0000	YES	YES	YES	YES	YES	YES	-	YES
DCC_REGS	DCC2_BASE	0x6008_1000	YES	YES	YES	YES	YES	YES	-	YES
DCC_REGS	DCC3_BASE	0x6008_2000	YES	YES	YES	YES	YES	YES	-	YES

12.5.2 DCC_REGS Registers

Table 12-3 lists the memory-mapped registers for the DCC_REGS registers. All register offset addresses not listed in Table 12-3 should be considered as reserved locations and the register contents should not be modified.

Table 12-3. DCC_REGS Registers

Offset	Acronym	Register Name	Protection
0h	DCCGCTRL	Starts / stops the counters. Clears the error signal.	
8h	DCCCNTSEED0	Seed value for the counter attached to Clock Source 0.	
Ch	DCCVALIDSEED0	Seed value for the timeout counter attached to Clock Source 0.	
10h	DCCCNTSEED1	Seed value for the counter attached to Clock Source 1.	
14h	DCCSTATUS	Specifies the status of the DCC Module.	
18h	DCCCNT0	Value of the counter attached to Clock Source 0.	
1Ch	DCCVALID0	Value of the valid counter attached to Clock Source 0.	
20h	DCCCNT1	Value of the counter attached to Clock Source 1.	
24h	DCCCLKSRC1	Selects the clock source for Counter 1.	
28h	DCCCLKSRC0	Selects the clock source for Counter 0.	

Complex bit access types are encoded to fit into small table cells. Table 12-4 shows the codes that are used for access types in this section.

Table 12-4. DCC_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
R-1	R -1	Read Returns 1s
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

12.5.2.1 DCCGCTRL Register (Offset = 0h) [Reset = 00005555h]

DCCGCTRL is shown in Figure 12-8 and described in Table 12-5.

Return to the Summary Table.

Starts / stops the counters. Clears the error signal.

Figure 12-8. DCCGCTRL Register

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

Table 12-5. DCCGCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-12	DONEENA	R/W	5h	DONE Enable Enables/disables the done interrupt signal, but has no effect on the done status flag in DCCSTAT register. 0101 The done signal is disabled Others The done signal is enabled Reset type: SYSRSn
11-8	SINGLESHOT	R/W	5h	Single-Shot Enable Enables/disables repetitive operation of the DCC. 1010: Stop counting when COUNTER0 and VALID0 both reach zero 1011: Reserved Others: Continuously repeat (until error) Reset type: SYSRSn
7-4	ERRENA	R/W	5h	Error Enable Enables/disables the error signal. 0101 The error signal is disabled Others The error signal is enabled Reset type: SYSRSn
3-0	DCCENA	R/W	5h	DCC Enable Starts and stops the operation of the DCC. 0101 Counters are stopped Others Counters are running Reset type: SYSRSn

12.5.2.2 DCCCNTSEED0 Register (Offset = 8h) [Reset = 00000000h]

DCCCNTSEED0 is shown in Figure 12-9 and described in Table 12-6.

Return to the Summary Table.

Seed value for the counter attached to Clock Source 0.

Figure 12-9. DCCCNTSEED0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												COUNTSEED0																			
R-0h												R/W-0h																			

Table 12-6. DCCCNTSEED0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19-0	COUNTSEED0	R/W	0h	Seed Value for Counter 0 Contains the seed value that gets loaded into Counter 0 (Clock Source 0). NOTE: Operating the DCC with '0' in the COUNTSEED0 register will result in undefined operation. Reset type: SYSRSn

12.5.2.3 DCCVALIDSEED0 Register (Offset = Ch) [Reset = 00000000h]

DCCVALIDSEED0 is shown in Figure 12-10 and described in Table 12-7.

Return to the Summary Table.

Seed value for the timeout counter attached to Clock Source 0.

Figure 12-10. DCCVALIDSEED0 Register

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

Table 12-7. DCCVALIDSEED0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	VALIDSEED	R/W	0h	Seed Value for Valid Duration Counter 0 Contains the seed value that gets loaded into the valid duration counter for Clock Source 0. NOTE: Operating the DCC with '0' in the VALIDSEED0 register will result in undefined operation. VALID0 defines a window in which COUNT1 expires. This window is meant to be at least four cycles wide. Do not program a value less than '4' into the VALID0 register. Reset type: SYSRSn

12.5.2.4 DCCCNTSEED1 Register (Offset = 10h) [Reset = 00000000h]

DCCCNTSEED1 is shown in Figure 12-11 and described in Table 12-8.

Return to the Summary Table.

Seed value for the counter attached to Clock Source 1.

Figure 12-11. DCCCNTSEED1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												COUNTSEED1																			
R-0h												R/W-0h																			

Table 12-8. DCCCNTSEED1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19-0	COUNTSEED1	R/W	0h	Seed Value for Counter 1 Contains the seed value that gets loaded into Counter 1 (Clock Source 1). NOTE: Operating the DCC with '0' in the COUNTSEED1 register will result in undefined operation. Reset type: SYSRSn

12.5.2.5 DCCSTATUS Register (Offset = 14h) [Reset = 00000000h]

DCCSTATUS is shown in Figure 12-12 and described in Table 12-9.

Return to the Summary Table.

Specifies the status of the DCC Module.

Figure 12-12. DCCSTATUS Register

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

Table 12-9. DCCSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	DONE	R/W	0h	Single-Shot Done Flag Indicates when single-shot mode is complete without error. Writing a '1' to this bit clears the flag. 0 Single-shot mode has not completed. 1 Single-shot mode has completed. Reset type: SYSRSn
0	ERR	R/W	0h	Error Flag Indicates whether or not an error has occurred. Writing a '1' to this bit clears the flag. 0 No errors have occurred. 1 An error has occurred. Reset type: SYSRSn

12.5.2.6 DCCCNT0 Register (Offset = 18h) [Reset = 00000000h]

DCCCNT0 is shown in Figure 12-13 and described in Table 12-10.

Return to the Summary Table.

Value of the counter attached to Clock Source 0.

Figure 12-13. DCCCNT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												COUNT0																			
R-0h												R-0h																			

Table 12-10. DCCCNT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19-0	COUNT0	R	0h	Current Value of Counter 0 Reset type: SYSRSn

12.5.2.7 DCCVALID0 Register (Offset = 1Ch) [Reset = 00000000h]

DCCVALID0 is shown in Figure 12-14 and described in Table 12-11.

Return to the Summary Table.

Value of the valid counter attached to Clock Source 0.

Figure 12-14. DCCVALID0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VALID0															
R-0h																R-0h															

Table 12-11. DCCVALID0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	VALID0	R	0h	Current Value of Valid 0 Reset type: SYSRSn

12.5.2.8 DCCCNT1 Register (Offset = 20h) [Reset = 00000000h]

DCCCNT1 is shown in Figure 12-15 and described in Table 12-12.

Return to the Summary Table.

Value of the counter attached to Clock Source 1.

Figure 12-15. DCCCNT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												COUNT1																			
R-0h												R-0h																			

Table 12-12. DCCCNT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19-0	COUNT1	R	0h	Current Value of Counter 1 Reset type: SYSRSn

12.5.2.9 DCCCLKSRC1 Register (Offset = 24h) [Reset = 00000000h]

DCCCLKSRC1 is shown in Figure 12-16 and described in Table 12-13.

Return to the Summary Table.

Selects the clock source for Counter 1.

Figure 12-16. DCCCLKSRC1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY				RESERVED						CLKSRC1					
R-0/W-0h				R-0h						R/W-0h					

Table 12-13. DCCCLKSRC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-12	KEY	R-0/W	0h	Enables or Disables Clock Source Write for COUNT1 1010 The CLKSRC field selects the clock source for COUNT1. Others: Previous values retained new writes on register fields has no impact. Reset type: SYSRSn
11-6	RESERVED	R	0h	Reserved

Table 12-13. DCCCLKSRC1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-0	CLKSRC1	R/W	0h	Clock Source Select for Counter 1 Specifies the clock source for COUNT1, when the KEY field enables this feature. Note: Any values not explicitly defined below are reserved. Reset type: SYSRSn 0h (R/W) = Direct output of SYSPLL CLKOUT 1h (R/W) = FCLK (divided clock) output from Flash Controller 1 2h (R/W) = INTOSC1 output clock 3h (R/W) = INTOSC2 output clock 4h (R/W) = Reserved 5h (R/W) = EtherCAT PHY clock 6h (R/W) = CPU1 clock 7h (R/W) = CPU2 clock. DCC monitors the point at which the clock to the primary and secondary modules of CPU2 lockstep implementation diverge in the LCM. 8h (R/W) = RTDMA clock. DCC monitors the point at which the clock to the primary and secondary modules of RTDMA lockstep implementation diverge in the LCM. 9h (R/W) = Input 15 of INPUTXBAR1 Ah (R/W) = Auxiliary clock input Bh (R/W) = Clock input to EPWM module Ch (R/W) = Reserved Dh (R/W) = ADC conversion clock Eh (R/W) = Watchdog clock after dividers Fh (R/W) = FCLK (divided clock) output from Flash Controller 2 (if present) 10h (R/W) = Reserved 11h (R/W) = Reserved 12h (R/W) = Reserved 13h (R/W) = Reserved 14h (R/W) = Reserved 15h (R/W) = CPU3 clock 16h (R/W) = Reserved 17h (R/W) = Reserved 18h (R/W) = Input 11 of INPUTXBAR1 19h (R/W) = Input 12 of INPUTXBAR1 1Ah (R/W) = MCANA bit clock 1Bh (R/W) = MCANB bit clock 1Ch (R/W) = MCANC bit clock 1Dh (R/W) = MCAND bit clock 1Eh (R/W) = MCANE bit clock 1Fh (R/W) = MCANF bit clock 20h (R/W) = ESM clock

12.5.2.10 DCCCLKSRC0 Register (Offset = 28h) [Reset = 00000000h]

DCCCLKSRC0 is shown in Figure 12-17 and described in Table 12-14.

Return to the Summary Table.

Selects the clock source for Counter 0.

Figure 12-17. DCCCLKSRC0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY				RESERVED							CLKSRC0				
R-0/W-0h				R-0h							R/W-0h				

Table 12-14. DCCCLKSRC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-12	KEY	R-0/W	0h	Enables or Disables Clock Source Write for COUNT0 1010: The CLKSRC0 field written with key gets updated to with new selection to clock COUNT0. Others: Previous values retained new writes on register fields has no impact. Reset type: SYSRSn
11-5	RESERVED	R	0h	Reserved
4-0	CLKSRC0	R/W	0h	Clock Source Select for Counter 0 Specifies the clock source for COUNT0, when the KEY field enables this feature. Note: All values not defined below are reserved. Reset type: SYSRSn 0h (R/W) = Crystal oscillator output 1h (R/W) = INTOSC1 output 2h (R/W) = INTOSC2 output 4h (R/W) = TCK pin input 5h (R/W) = CPU1 clock 8h (R/W) = Auxiliary clock input Ch (R/W) = Input 16 of INPUTXBAR1 Eh (R/W) = Reserved Fh (R/W) = Reserved

Chapter 13

Real-Time Direct Memory Access (RTDMA)

The Real-Time Direct Memory Access (RTDMA) module provides a hardware method of transferring data between peripherals and memory without intervention from the CPU; thereby, freeing up bandwidth for other system functions. Additionally, the RTDMA has the capability to orthogonally rearrange the data as the data is transferred as well as “ping-pong” data between buffers. These features are useful for structuring data into blocks for CPU processing.

13.1 Introduction

The strength of a controller is not measured purely in processor speed, but in total system capabilities. As a part of the equation, any time the CPU bandwidth for a given function can be reduced, the greater the system capabilities. Many times applications spend a significant amount of their bandwidth moving data, whether moving data from off-chip memory to on-chip memory, from a peripheral such as an analog-to-digital converter (ADC) to RAM, or from one peripheral to another. Furthermore, many times this data comes in a format that is not conducive to the optimal processing powers of the CPU. The RTDMA module described in this chapter has the ability to free up CPU bandwidth and rearrange the data into a pattern for more streamlined processing in real time.

The RTDMA module is an event-based machine, meaning the RTDMA module requires a peripheral, channel, or software trigger to start a RTDMA transfer. The RTDMA module can be made into a periodic time-driven machine by configuring a timer as the RTDMA trigger source as well as utilizing the channels within the module itself to start memory transfers periodically. The RTDMA module has ten independent RTDMA channels that can be configured separately, and each channel contains their own independent Interrupt Controller interrupt to let the CPU know when a RTDMA transfer has either started or completed. All ten channels can be configured at one of four priority levels with one selected channel at a higher priority than the others. At the heart of the RTDMA is a state machine and tightly coupled address control logic. This address control logic allows for rearrangement of the block of data during the transfer as well as the process of ping-ponging data between buffers. Each of these features is discussed in detail in this chapter.

13.1.1 Features

RTDMA features include:

- 10 RTDMA channels with software configurable priority levels and independent Interrupt Controller interrupts
- Up to 256 hardware trigger sources to initiate RTDMA transfers
- Internal trigger generation for data transfers and trigger sources for channels
- Independent Read and Write buses
- Word Size: 8-bit, 16-bit, 32-bit, and 64-bit transfers
- Throughput: 1 cycle/word after the initial read-write access with 0 cycle read/write stall
- FIFO implemented within hardware to optimize data transfers
- Linear and circular addressing modes
- Support for multiple data transformation functions as data is transferred from source to destination
  - Ability to reverse words, half words, and so on.
- Burst Mode Support (for transfers with EMIF)
- Access protection through the Memory Protection Unit (MPU)

13.1.2 RTDMA Related Collateral

Foundational Materials

- C29x Academy - RTDMA Lab
- C29x Academy - Real-Time Direct Memory Access (RTDMA)

13.1.3 Block Diagram

Figure 13-1 shows the block diagram of the RTDMA.

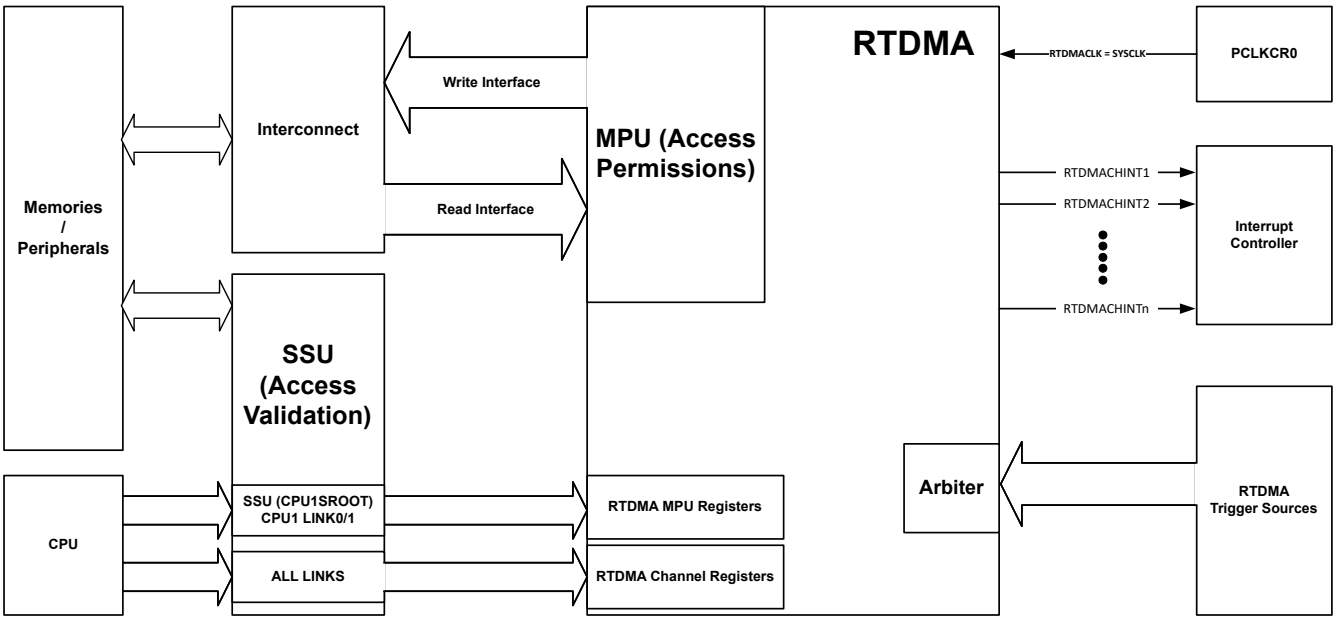


Figure 13-1. RTDMA Block Diagram

13.2 RTDMA Trigger Source Options

Table 13-1 lists the RTDMA trigger source options.

Table 13-1. RTDMA Trigger Source Options

Select Index	Trigger Source
0	DMA_SOFTWARE_TRIGGER
1	ADCAINT1_DMA
2	ADCAINT2_DMA
3	ADCAINT3_DMA
4	ADCAINT4_DMA
5	ADCAEVT
6	ADCBINT1_DMA
7	ADCBINT2_DMA
8	ADCBINT3_DMA
9	ADCBINT4_DMA
10	ADCBEVT
11	ADCCINT1_DMA
12	ADCCINT2_DMA
13	ADCCINT3_DMA
14	ADCCINT4_DMA
15	ADCCEVT
16	ADCDINT1_DMA
17	ADCDINT2_DMA
18	ADCDINT3_DMA
19	ADCDINT4_DMA

Table 13-1. RTDMA Trigger Source Options (continued)

Select Index	Trigger Source
20	ADCDEVT
21	ADCEINT1_DMA
22	ADCEINT2_DMA
23	ADCEINT3_DMA
24	ADCEINT4_DMA
25	ADCEEVT
26-30	Reserved
31	CPU1_XINT1
32	CPU1_XINT2
33	CPU1_XINT3
34	CPU1_XINT4
35	CPU1_XINT5
36	CPU2_XINT1
37	CPU2_XINT2
38	CPU2_XINT3
39	CPU2_XINT4
40	CPU2_XINT5
41	CPU3_XINT1
42	CPU3_XINT2
43	CPU3_XINT3
44	CPU3_XINT4
45	CPU3_XINT5
46	EPWM1_SOCA
47	EPWM1_SOCB
48	EPWM2_SOCA
49	EPWM2_SOCB
50	EPWM3_SOCA
51	EPWM3_SOCB
52	EPWM4_SOCA
53	EPWM4_SOCB
54	EPWM5_SOCA
55	EPWM5_SOCB
56	EPWM6_SOCA
57	EPWM6_SOCB
58	EPWM7_SOCA
59	EPWM7_SOCB
60	EPWM8_SOCA
61	EPWM8_SOCB
62	EPWM9_SOCA
63	EPWM9_SOCB
64	EPWM10_SOCA
65	EPWM10_SOCB
66	EPWM11_SOCA
67	EPWM11_SOCB
68	EPWM12_SOCA

Table 13-1. RTDMA Trigger Source Options (continued)

Select Index	Trigger Source
69	EPWM12_SOCB
70	EPWM13_SOCA
71	EPWM13_SOCB
72	EPWM14_SOCA
73	EPWM14_SOCB
74	EPWM15_SOCA
75	EPWM15_SOCB
76	EPWM16_SOCA
77	EPWM16_SOCB
78	EPWM17_SOCA
79	EPWM17_SOCB
80	EPWM18_SOCA
81	EPWM18_SOCB
82-93	Reserved
94	CPU1_TINT0
95	CPU1_TINT1
96	CPU1_TINT2
97	CPU2_TINT0
98	CPU2_TINT1
99	CPU2_TINT2
100	CPU3_TINT0
101	CPU3_TINT1
102	CPU3_TINT2
103-111	Reserved
112	ECAP1_DMA
113	ECAP2_DMA
114	ECAP3_DMA
115	ECAP4_DMA
116	ECAP5_DMA
117	ECAP6_DMA
118-119	Reserved
120	LINA_TXDMA
121	LINA_RXDMA
122	LINB_TXDMA
123	LINB_RXDMA
124	ECAT_SYNC_DMA_TRIG
125	SPIA_TXDMA
126	SPIA_RXDMA
127	SPIB_TXDMA
128	SPIB_RXDMA
129	SPIC_TXDMA
130	SPIC_RXDMA
131	SPID_TXDMA
132	SPID_RXDMA
133	SPIE_TXDMA

Table 13-1. RTDMA Trigger Source Options (continued)

Select Index	Trigger Source
134	SPIE_RXDMA
135	CLB1_INT
136	CLB2_INT
137	CLB3_INT
138	CLB4_INT
139	CLB5_INT
140	CLB6_INT
141-142	Reserved
143	FSITXA_DMA
144	FSIRXA_DMA
145	FSIRXA_DATA_TAG_MATCH
146	FSIRXA_PING_TAG_MATCH
147	FSITXB_DMA
148	FSIRXB_DMA
149	FSIRXB_DATA_TAG_MATCH
150	FSIRXB_PING_TAG_MATCH
151	FSITXC_DMA
152	FSIRXC_DMA
153	FSIRXC_DATA_TAG_MATCH
154	FSIRXC_PING_TAG_MATCH
155	FSITXD_DMA
156	FSIRXD_DMA
157	FSIRXD_DATA_TAG_MATCH
158	FSIRXD_PING_TAG_MATCH
159-160	Reserved
161	CPU1_DLT
162	CPU2_DLT
163	CPU3_DLT
164-166	Reserved
167	UARTA_RX
168	UARTA_TX
169	UARTB_TX
170	UARTB_RX
171	UARTC_TX
172	UARTC_RX
173	UARTD_TX
174	UARTD_RX
175	UARTE_TX
176	UARTE_RX
177	UARTF_TX
178	UARTF_RX
179	DTHE_SHA_DMA_S_CTXIN_REQ
180	DTHE_SHA_DMA_S_DATAIN_REQ
181	DTHE_SHA_DMA_S_CTXOUT_REQ
182	DTHE_SHA_DMA_P_CTXIN_REQ

Table 13-1. RTDMA Trigger Source Options (continued)

Select Index	Trigger Source
183	DTHE_SHA_DMA_P_DATAIN_REQ
184	DTHE_SHA_DMA_P_CTXOUT_REQ
185	DTHE_AES_DMA_S_CTXIN_REQ
186	DTHE_AES_DMA_S_DATAIN_REQ
187	DTHE_AES_DMA_S_DATAOUT_REQ
188	DTHE_AES_DMA_S_CTXOUT_REQ
189	DTHE_AES_DMA_P_CTXIN_REQ
190	DTHE_AES_DMA_P_DATAIN_REQ
191	DTHE_AES_DMA_P_DATAOUT_REQ
192	DTHE_AES_DMA_P_CTXOUT_REQ
193	DTHE_SM3_CTXIN_REQ
194	DTHE_SM3_DATAIN_REQ
195	DTHE_SM3_CTXOUT_REQ
196	DTHE_SM4_ CTXIN_REQ
197	DTHE_SM4_ DATAIN_REQ
198	DTHE_SM4_ DATAOUT_REQ
199	DTHE_SM4_ CTXOUT_REQ
200	EPG_INT
201	SD1FLT1_DRINT
202	SD1FLT2_DRINT
203	SD1FLT3_DRINT
204	SD1FLT4_DRINT
205	SD2FLT1_DRINT
206	SD2FLT2_DRINT
207	SD2FLT3_DRINT
208	SD2FLT4_DRINT
209	SD3FLT1_DRINT
210	SD3FLT2_DRINT
211	SD3FLT3_DRINT
212	SD3FLT4_DRINT
213	SD4FLT1_DRINT
214	SD4FLT2_DRINT
215	SD4FLT3_DRINT
216	SD4FLT4_DRINT
217	SENT1
218	SENT2
219	SENT3
220	SENT4
221	SENT5
222	SENT6
223	WADI1_DMAREQ
224	WADI2_DMAREQ
225-239	Reserved
240	RTDMA_CH1INT
241	RTDMA_CH2INT

Table 13-1. RTDMA Trigger Source Options (continued)

Select Index	Trigger Source
242	RTDMA_CH3INT
243	RTDMA_CH4INT
244	RTDMA_CH5INT
245	RTDMA_CH6INT
246	RTDMA_CH7INT
247	RTDMA_CH8INT
248	RTDMA_CH9INT
249	RTDMA_CH10INT
250-255	Reserved

13.3 RTDMA Bus

The RTDMA bus architecture consists of a 32-bit address bus, a 64-bit data read bus, and a 64-bit data write bus. Memories and register locations connected to the RTDMA bus by way of interfaces that sometimes share resources with the CPU memory or peripheral bus.

Note

RTDMA allows RD and WR ports to have different size configurations. DATASIZE (of the RD port) must always be configured as an integral multiple of WR_DATASIZE configuration. It is still recommended to keep the RD and WR ports to have the same size configurations.

- EX: If DATASIZE_64, WR_DATASIZE can be configured as WR_DATASIZE_64, WR_DATASIZE_32, WR_DATASIZE_16, or WR_DATASIZE_8.

Refer to the device memory map for the complete list of allowed peripheral and memory accesses with the RTDMA.

13.4 Address Pointer and Transfer Control

The RTDMA state machine is, at the most basic level, two nested loops.

Burst (Inner) Loop:

The burst (inner) loop transfers a programmable number of bytes set by (BURST_SIZE + 1) register when a RTDMA channel trigger (Peripheral or Software trigger) is received. The BURST_SIZE register allows a maximum of 256 bytes to be transferred in one burst. Each RTDMA channel supports 8-bit, 16-bit, 32-bit, or 64-bit bursts that can be controlled by MODE.DATASIZE bit field. Each RTDMA channel contains a shadowed address pointer for the source (SRC_ADDR_SHADOW) and the destination (DST_ADDR_SHADOW) address. At the beginning of each transfer, the shadowed version of each pointer is copied into the respective active (SRC_ADDR_ACTIVE or DST_ADDR_ACTIVE) register. During the burst loop, after each byte is transferred, the signed value contained in the appropriate source or destination BURST_STEP register is added to the active register:

SRC_ADDR_ACTIVE = SRC_ADDR_ACTIVE + SRC_BURST_STEP

DST_ADDR_ACTIVE = DST_ADDR_ACTIVE + DST_BURST_STEP

The burst (inner) loop transfers a burst of data when a RTDMA Channel Trigger (Peripheral or Software trigger) is received.

Transfer (Outer) Loop:

The Transfer (outer) loop transfers a programmable number of bursts set by (TRANSFER_SIZE + 1) register for each channel. Since TRANSFER_SIZE is a 32-bit register, the total size of a transfer allowed is well beyond any practical requirement. During the transfer loop, after each burst is complete, there are two methods that can be used to modify the active address pointer.

Method 1 (Default): When address wrapping is disabled (SRC_WRAP_SIZE or DST_WRAP_SIZE is greater than TRANSFER_SIZE), active address pointer is updated as shown below

SRC_ADDR_ACTIVE = SRC_ADDR_ACTIVE + SRC_TRANSFER_STEP

DST_ADDR_ACTIVE = DST_ADDR_ACTIVE + DST_TRANSFER_STEP

Method 2: Address wrapping gets enabled when SRC_WRAP_SIZE or DST_WRAP_SIZE is less than TRANSFER_SIZE. This allows the channel to wrap multiple times within a single transfer. When the number of bursts is equal to (SRC/DST_WRAP_SIZE + 1) register, the state machine modifies the active address pointers as:

SRC_BEG_ADDR_ACTIVE = SRC_BEG_ADDR_ACTIVE + SRC_WRAP_STEP

DST_BEG_ADDR_ACTIVE = DST_BEG_ADDR_ACTIVE + DST_WRAP_STEP

SRC_ADDR_ACTIVE = SRC_BEG_ADDR_ACTIVE

DST_ADDR_ACTIVE = DST_BEG_ADDR_ACTIVE

At the end of RTDMA transfer, RTDMA can have transferred (BURST_SIZE + 1) x (TRANSFER_SIZE + 1) words.

Note

The RD and WR ports can be configured with different data sizes, make sure reads and writes are happening properly.

OneShot Mode:

Note

The OneShot mode can create a condition where one trigger uses up the majority of the RTDMA bandwidth and causes long CPU stalls. It is recommended to configure a CPU timer (or similar) and disable ONESHOT to avoid this situation.

The OneShot mode must not be used with channels configured with a priority of 0.

OneShot mode is disabled by default.

When OneShot mode is disabled (MODE.CHx[ONESHOT] = 0), RTDMA transfers one burst [(BURST_SIZE + 1) bytes] of data each time a RTDMA Channel Trigger is received until (TRANSFER_COUNT = 0). After the burst is completed, the state machine moves on to the next pending channel in the priority scheme, even if another trigger for the channel just completed is pending. This feature keeps any single channel from monopolizing the RTDMA bus.

When OneShot mode is enabled (MODE.CHx[ONESHOT] = 1), RTDMA transfers all the bursts [(BURST_SIZE + 1) x (TRANSFER_SIZE + 1) words] on a single RTDMA channel trigger until (TRANSFER_COUNT = 0).

Continuous Mode:

Continuous mode is disabled by default.

When Continuous mode is disabled (MODE.CHx[CONTINUOUS] = 0), RTDMA state machine disables channel after all bursts in a transfer loop (TRANSFER_COUNT = 0) are complete. The channel stops and the RUNSTS bit is cleared. The channel must be re-enabled by setting the RUN bit in the CONTROL register before another transfer can be started on that channel.

When Continuous mode is enabled (MODE.CHx[CONTINUOUS] = 1), RTDMA state machine keep channel active even after all bursts in a transfer loop (TRANSFER_COUNT = 0) are complete.

Each RTDMA channel can trigger an Interrupt Controller interrupt for each RTDMA transfer either at start of RTDMA transfer or end of RTDMA transfer using MODE.CHx[CHINTMODE] bit.

When an RTDMA channel receives a peripheral event trigger, the PERINTFLG bit in the CONTROL register is set, pending enabling of the channel to the RTDMA state machine. To initiate an RTDMA event using software, setting the PERINTFRC bit forces an event.

Source/Destination Address Pointers (SRC/DST_ADDR)

The value written into the shadow register is the start address of the first location where data is read or written to.

At the beginning of a transfer the shadow register (SRC/DST_ADDR_SHADOW) is copied into the active register (SRC/DST_ADDR_ACTIVE). The active register performs as the current address pointer.

Source/Destination Begin Address Pointers (SRC/DST_BEG_ADDR)

This is the wrap pointer.

The value written into the shadow register (SRC/DST_BEG_ADDR_SHADOW) is loaded into the active register (SRC/DST_BEG_ADDR_ACTIVE) at the start of a transfer. On a wrap condition, the active register (SRC/DST_BEG_ADDR_ACTIVE) is incremented by the signed value in the appropriate SRC/DST_WRAP_STEP register prior to being loaded into the active register ((SRC/DST_ADDR_ACTIVE).

For each channel, the transfer process can be controlled with the following size values:

Source and Destination Burst Size (BURST_SIZE)

This specifies the number of bytes to be transferred in a burst.

This value is loaded into the BURST_COUNT register at the beginning of each burst. The BURST_COUNT decrements each byte that is transferred and when the register reaches a zero value, the burst is complete, indicating that the next channel can be serviced. The behavior of the current channel is defined by the ONE_SHOT bit in the MODE register. The maximum size of the burst is dictated by the type of peripheral. For the ADC, the burst size can be all 32 registers (if all 32 registers are used). For RAM, the burst size can be up to the maximum allowed by the BURST_SIZE register, which is 256. See Table 13-2 to understand how BURST_SIZE register affects the number of bytes transferred with respect to DATASIZE.

Table 13-2. BURSTSIZE versus DATASIZE Behavior

<table>
<tr><th rowspan="2">BURSTSIZE</th><th colspan="4">Number of Bytes Transferred in</th></tr>
<tr><th>DATASIZE = 8-bit</th><th>DATASIZE = 16-bit</th><th>DATASIZE = 32-bit</th><th>DATASIZE = 64-bit</th></tr>
<tr><td>0</td><td>1</td><td rowspan="2">2</td><td rowspan="4">4</td><td rowspan="8">8</td></tr>
<tr><td>1</td><td>2</td></tr>
<tr><td>2</td><td>3</td><td rowspan="2">4</td></tr>
<tr><td>3</td><td>4</td></tr>
<tr><td>4</td><td>5</td><td rowspan="2">6</td><td rowspan="4">8</td></tr>
<tr><td>5</td><td>6</td></tr>
<tr><td>6</td><td>7</td><td rowspan="2">8</td></tr>
<tr><td>7</td><td>8</td></tr>
<tr><td>8</td><td>9</td><td rowspan="2">10</td><td rowspan="4">12</td><td rowspan="4">16</td></tr>
<tr><td>9</td><td>10</td></tr>
<tr><td>10</td><td>11</td><td rowspan="2">12</td></tr>
<tr><td>11</td><td>12</td></tr>
<tr><td>*
*
*
*
*
*
</td><td>
*
*
*
*
*
</td><td>
*
*
*
*
255</td><td>*
255
256</td><td>*
256</td></tr>
<tr><td>254</td><td>255</td><td rowspan="2">256</td><td rowspan="2">256</td></tr>
<tr><td>255</td><td>256</td></tr>
</table>

In other words, BURSTSIZE has to be configured as a multiple of DATASIZE. The following configurations are the only allowed options:

	DATASIZE = 8-bit n = 1	DATASIZE = 16-bit n = 2	DATASIZE = 32-bit n = 4	DATASIZE = 64-bit n = 8
BURSTSIZE	n - 1	2n - 1	4n - 1	8n - 1

Source and Destination Transfer Size (TRANSFER_SIZE)

This specifies the number of bursts to be transferred per CPU interrupt (if enabled).

Whether this interrupt is generated at the beginning or the end of the transfer is defined in the CHINTMODE bit in the MODE register. Whether the channel remains enabled or not after the transfer is completed is defined by the CONTINUOUS bit in the MODE register. The TRANSFER_SIZE register is loaded into the TRANSFER_COUNT register at the beginning of each transfer. The TRANSFER_COUNT register keeps track of how many bursts of data the channel has transferred and when the register reaches zero, the RTDMA transfer is complete.

Source/Destination Wrap Size (SRC/DST_WRAP_SIZE)

This specifies the number of bursts to be transferred before the current address pointer wraps around to the beginning.

This feature is used to implement a circular addressing type function. This value is loaded into the appropriate SRC/DST_WRAP_COUNT register at the beginning of each transfer. The SRC/DST_WRAP_COUNT registers keep track of how many bursts of data the channel has transferred and when the registers reach zero, the wrap procedure is performed on the appropriate source or destination address pointer. A separate size and count register is allocated for source and destination pointers. To *disable* the wrap function, assign the value of these registers to be larger than the TRANSFER_SIZE.

Note

The values written to the BURST_SIZE and TRANSFER_SIZE registers are one less than the intended size. To transfer three 8-bit words in one transfer, the value 2 can be placed in the BURST_SIZE and the value 0 can be placed in the TRANSFER_SIZE registers. To transfer three 16-bit words in one transfer, the value 5 can be placed in the BURST_SIZE and the value 0 can be placed in the TRANSFER_SIZE registers.

Regardless of the state of the DATASIZE bit, the value specified in the STEP registers are for 8-bit addresses. So, to transfer one 32-bit word, a value 4 can be placed in these registers.

For each source/destination pointer, the address changes can be controlled with the following step values:

Source/Destination Burst Step (SRC/DST_BURST_STEP)

Within each burst transfer, the address source and destination step sizes are specified by these registers.

This value is a signed 2s compliment number so that the address pointer can be incremented or decremented as required. If no increment is desired, such as when accessing the data receive or transmit registers in a communication peripheral, the value of these registers can be set to zero.

Source/Destination Transfer Step (SRC/DST_TRANSFER_STEP)

This specifies the address offset to start the next burst transfer after completing the current burst transfer.

This is used in cases where registers or data memory locations are spaced at constant intervals. This value is a signed 2s compliment number so that the address pointer can be incremented or decremented as required.

Source/Destination Wrap Step (SRC/DST_WRAP_STEP)

When the wrap counter reaches zero, this value specifies the number of words to add/subtract from the SRC/DST_BEG_ADDR pointer and hence sets the new start address.

This implements a circular type of addressing mode, useful in many applications. This value is a signed 2s compliment number so that the address pointer can be incremented or decremented as required.

Note

Regardless of the state of the DATASIZE bit, the value specified in the STEP registers are for 16-bit addresses. So, to increment one 32-bit address, a value of 4 can be placed in these registers.

Channel Interrupt Mode (CHINTMODE)

This mode bit selects whether the RTDMA interrupt from the respective channel is generated at the beginning of a new transfer or at the end of the transfer.

If implementing a ping-pong buffer scheme with continuous mode of operation, then the interrupt can be generated at the beginning, just after the working registers are copied to the shadow set. If the RTDMA does not operate in continuous mode, then the interrupt is typically generated at the end when the transfer is complete.

All of the previous features and modes are shown in Figure 13-2. The following items are in reference to Figure 13-2.

- The *HALT* points represent where the channel halts operation when interrupted by a high priority channel trigger, or when the HALT command is set, or when an emulation halt is issued and the FREE bit is cleared to 0.
- The SRC/DST_ADDR_ACTIVE registers are not affected by SRC/DST_BEG_ADDR_ACTIVE at the start of a transfer. SRC/DST_BEG_ADDR_ACTIVE only affects the SRC/DST_ADDR_ACTIVE registers on a wrap. Following is what happens when a transfer first starts:
  - SRC/DST_BEG_ADDR_SHADOW remains unchanged.
  - SRC/DST_ADDR_SHADOW remains unchanged.
  - SRC/DST_BEG_ADDR_ACTIVE = SRC/DST_BEG_ADDR_SHADOW
  - SRC/DST_ADDR_ACTIVE = SRC/DST_ADDR_SHADOW
- The active registers get updated when a wrap occurs. The shadow registers remain unchanged. Specifically:
  - SRC/DST_BEG_ADDR_SHADOW remains unchanged.
  - SRC/DST_ADDR_SHADOW remains unchanged.
  - SRC/DST_BEG_ADDR_ACTIVE += SRC/DST_WRAP_STEP
  - SRC/DST_ADDR_ACTIVE = SRC/DST_BEG_ADDR_ACTIVE
- The best way to remember this is:
  - The shadow registers never change except by software.
  - The active registers never change except by hardware, and a shadow register is only copied into the active register, never an active register by another name.

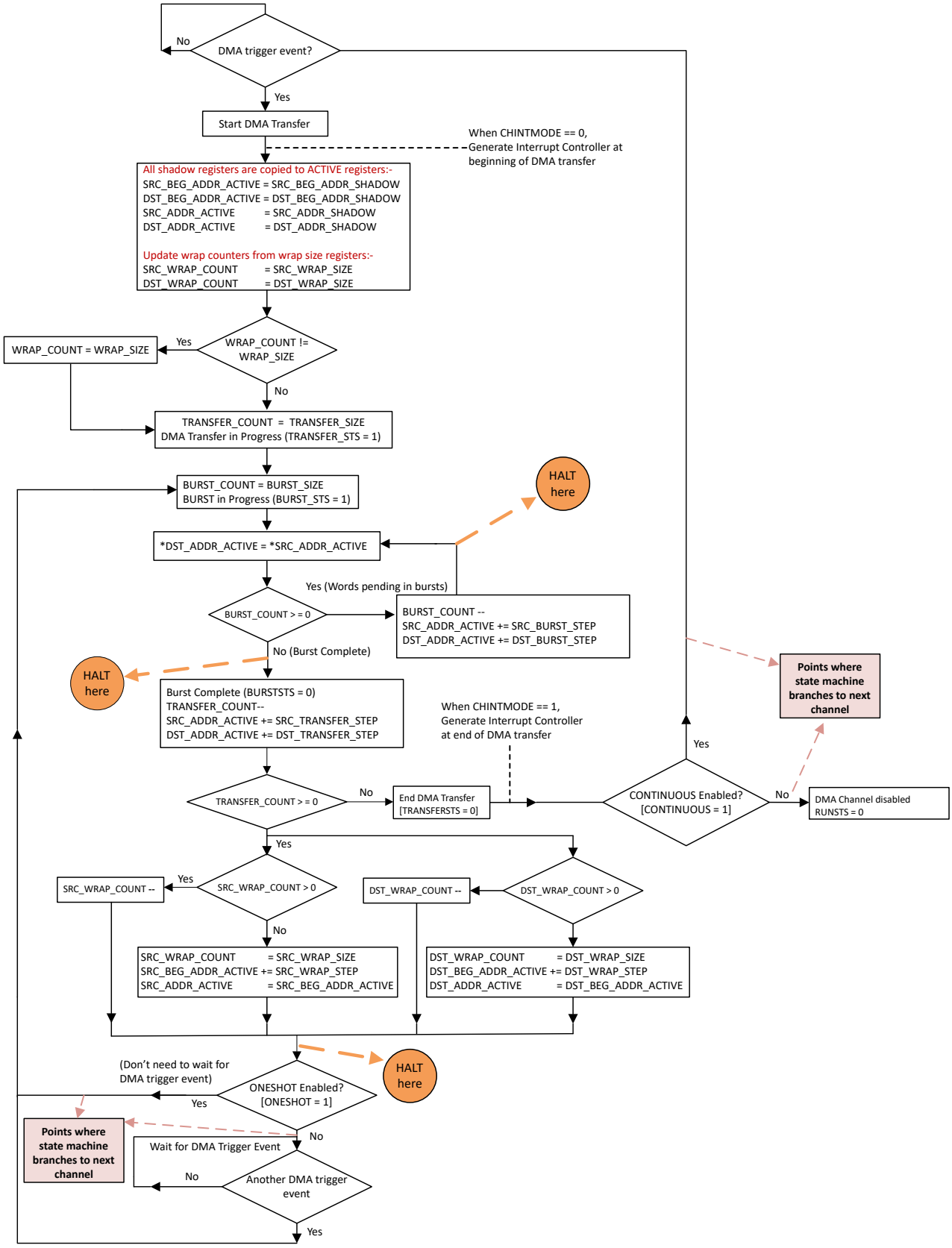


Figure 13-2. RTDMA State Diagram

13.5 Pipeline Timing and Throughput

In addition to the pipeline there are a few other behaviors of the RTDMA that affect the total throughput:

- A 1-cycle delay is added between bursts of different channels.
- A 1-cycle delay is added when a burst is interrupted by a channel with priority equal to 0.
- A 1-cycle delay is added when a burst is interrupted by a CPU suspend or the HALT bit.
- The last cycle does not require an extra cycle delay.
- 32-bit transfers run at double the speed of a 16-bit transfer (takes the same amount of time to transfer a 32-bit word as to transfer a 16-bit word).

For example, to transfer 128 16-bit words from LDA0 RAM to LDA3 RAM, a channel can be configured to transfer 8 bursts of 16 words/burst. The transfer can take:

1 burst × 4 cycles/word + 7 bursts × [(16 cycles/word × 1 words/burst + 1 cycles/word × 1 words/burst) + 1] = 139 cycles

If instead the channel were configured to transfer the same amount of data 32 bits at a time (the word size is configured to 32 bits), the transfer can take:

1 burst × 4 cycles/word + 7 bursts × [(8 cycles/word × 1 words/burst + 1 cycles/word × 1 words/burst) + 1] = 139 cycles

The RTDMA module consists of a 3-stage pipeline as shown in Figure 13-3, Figure 13-4, Figure 13-5, and Figure 13-6.

Note

These diagrams refer to peripherals that are edge triggered. For peripherals that are level triggered, the next arbitration happens only after the last write of the previous burst is complete.

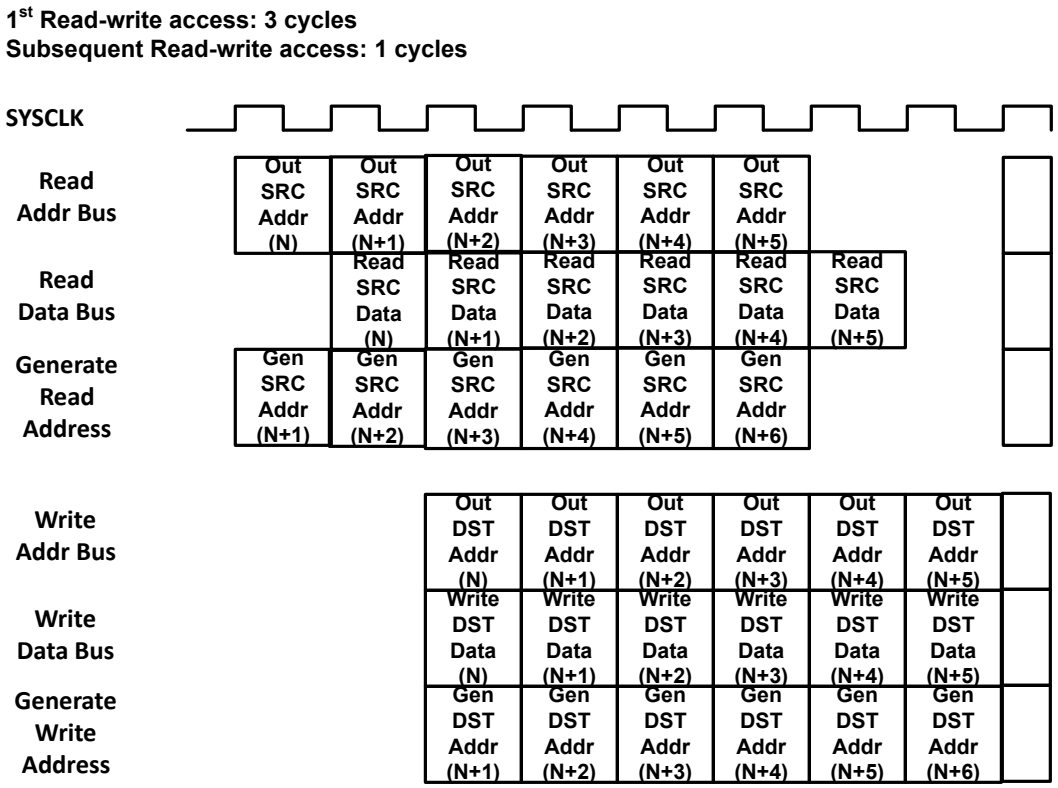


Figure 13-3. 3-stage Pipeline: With 0 Cycle Read Stall and 0 Cycle Write Stall

1st Read-write access: 5 cycles
Subsequent Read-write access: 2 cycles

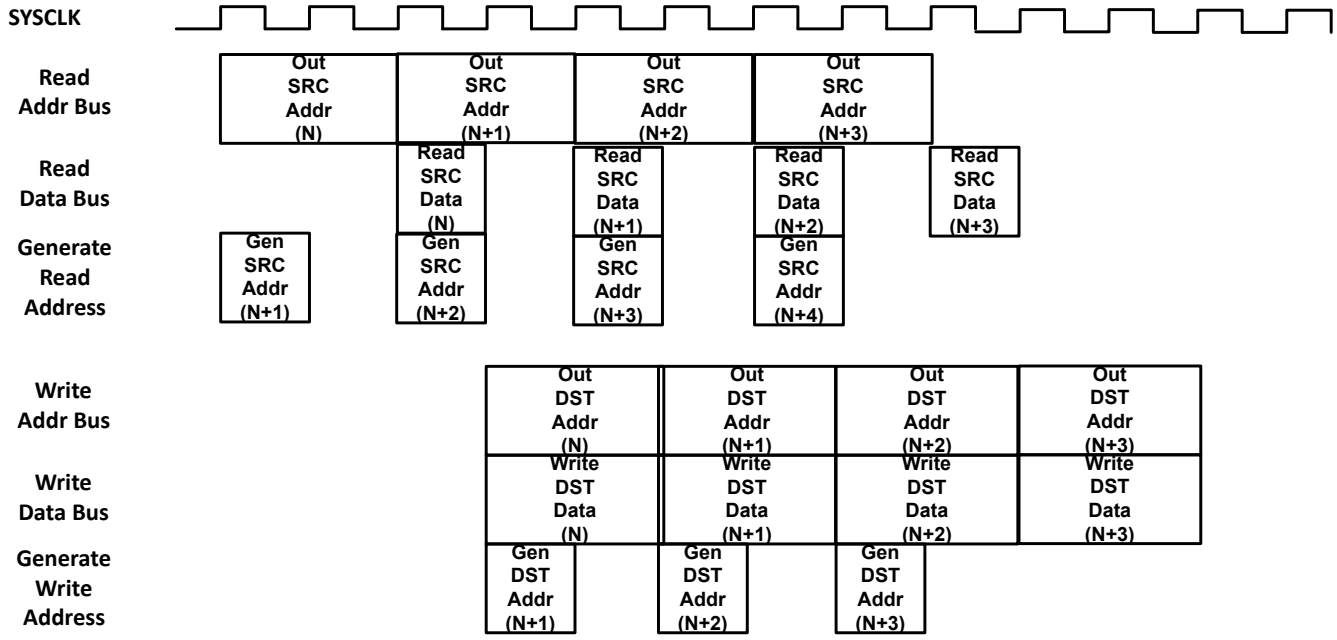


Figure 13-4. 3-stage Pipeline: With One Read Stall

1st Read-write access: 4 cycles
Back to back Read accesses, Write accesses from FIFO

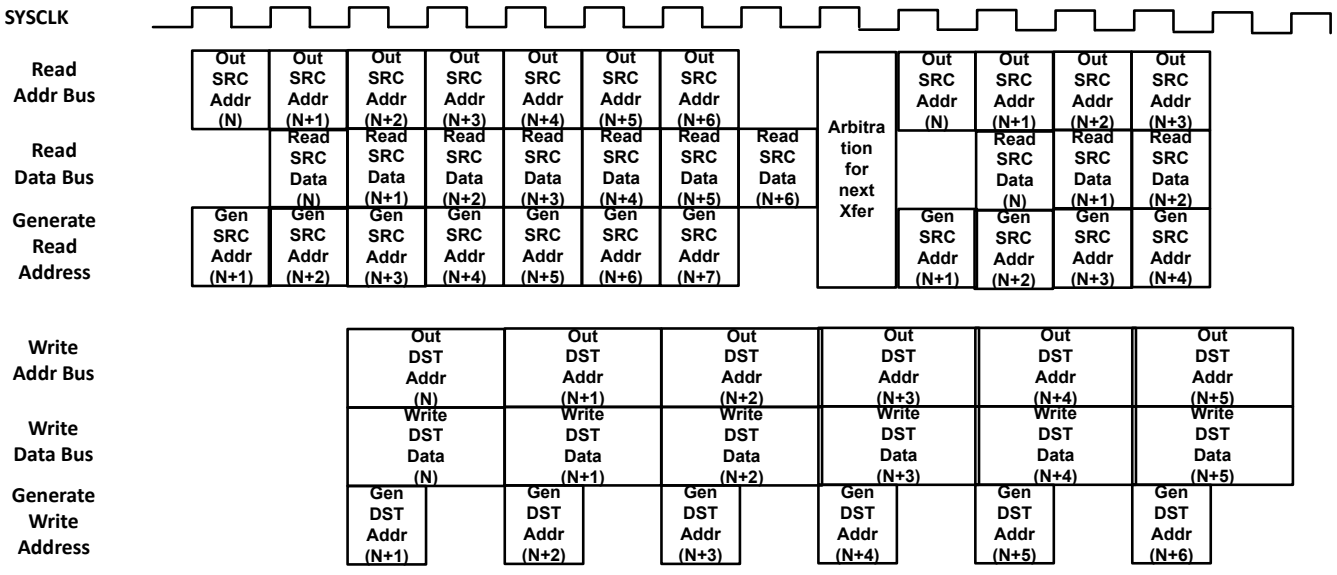


Figure 13-5. 3-stage Pipeline: With One Write Stall

1st Read-write access: 7 cycles
Subsequent Read-write access: Depending on the wait states of read/write accesses
Back to back read accesses, Write accesses from FIFO

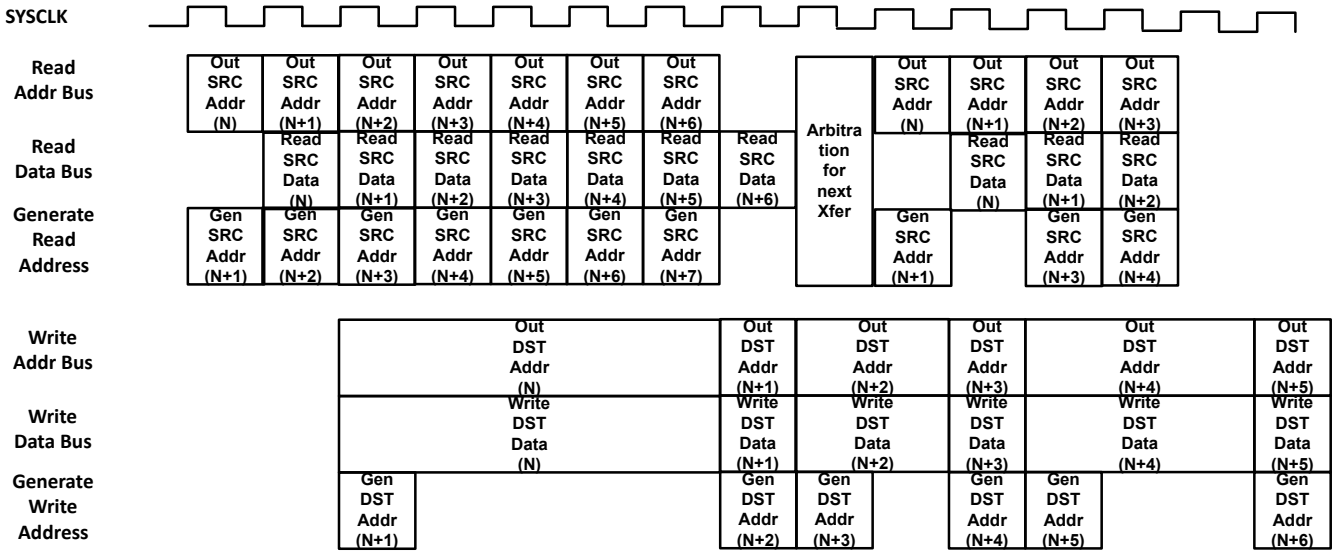


Figure 13-6. 3-stage Pipeline: With Multicycle Write Stall

13.6 Channel Priority

The priority of the RTDMA channels is configurable in two modes:

13.6.1 Round-Robin Mode

In this mode, all channels have *equal* priority and each enabled channel is serviced in round-robin fashion as follows:

CH1 → CH2 → CH3 → CH4 → CH5 → ... → CH10 → CH1 → CH2 → …

In the case above, after each channel has transferred a burst of bytes, the next channel is serviced. The user can specify the size of the burst for each channel. Once CH10 (or the last enabled channel) has been serviced, and no other channels are pending, the round-robin state machine remains in the current state. The round-robin state machine enters an idle state only when there a priority, hard, or system reset is executed.

From the idle state, channel 1 (if enabled) is always serviced first. However, if the RTDMA is currently processing another channel x, all other pending channels between x and the end of the round are serviced before CH1. All the channels are of *equal* priority. For instance, take an example where CH1, CH4, and CH5 are enabled in round-robin mode and CH4 is currently being processed. Then CH1 and CH5 both receive an interrupt trigger from their respective peripherals before CH4 completes. CH1 and CH5 are now both pending. When CH4 completes the burst, CH5 is serviced next. Only after CH5 completes is CH1 serviced. Upon completion of CH1, if there are no more channels pending, the round-robin state machine enters an idle state.

A more complicated example is:

- Assume all channels are enabled, and the RTDMA is in an idle state,
- Initially a trigger occurs on CH1, CH3, and CH5 on the same cycle,
- When the CH1 burst transfer starts, requests from CH3 and CH5 are pending,
- Before completion of the CH1 burst, the RTDMA receives a request from CH2. Now the pending requests are from CH2, CH3, and CH5,
- After completing the CH1 burst, CH2 is serviced since this channel is next in the round-robin scheme after CH1.
- After the burst from CH2 is finished, the CH3 burst is serviced, followed by CH5 burst.
- Now while the CH5 burst is being serviced, the RTDMA receives a request from CH1, CH3, and CH6.
- The burst from CH6 starts after the completion of the CH5 burst, since this channel is the next channel after CH5 in the round-robin scheme.
- This is followed by the CH1 burst and then the CH3 burst
- After the CH3 burst finishes, assuming no more triggers have occurred, the round-robin state machine enters an idle state.

The round-robin state machine can be reset to the idle state using the DMACTRL[PRIORITYRESET] bit.

Note

It is recommended to only modify the PRIORITYRESET bit in a reset sequence.

13.6.2 Software Configurable Priority of Channels

In this mode, each of the 10 channels can be set to a software configurable priority of 0 to 3 using the registers SWPRI1 (for channels 1-8) and SWPRI2 (for channels 9-10). The channels can be assigned priority orthogonally, that is, any channel can be set to any priority. The lowest priority number implies highest priority during arbitration. When the same priority number is assigned to multiple channels, during the arbitration, the channel with lowest channel number assigned wins the arbitration. All the channels are set to a default priority of 1.

A channel with PRI = 0 is considered as a special case. In this mode, if a channel with PRI = 0 event occurs, the current word transfer (read then write) on any other channel is completed (not the complete burst) and execution is halted. The channel with PRI = 0 is then serviced for the complete BURST count. When this channel with PRI = 0 burst is complete, execution returns to the channel that was active when the channel with PRI = 0 event occurred. The configuration listed below is one use case of how priority levels can be used to enhance RTDMA operation.

Higher priority:	CH5 (PRI = 0)
Lower priority:	CH3 (PRI = 1) → CH12(PRI = 2) → CH1(PRI = 3 → CH15(PRI = 4) → CH6(PRI = 5) → CH2(PRI = 6) → …

Multiple channels can be configured to be PRI = 0. Another example is when CH1, CH4, and CH5 are enabled in Channel 1 high-priority mode and CH4 is currently being processed. CH1 and CH5 both receive an interrupt trigger from the respective peripherals before CH4 completes. CH1 and CH5 are now both pending. When the current CH4 word transfer is completed, regardless of whether the RTDMA has completed the entire CH4 burst, CH4 execution is suspended and CH1 is serviced. After the CH1 burst completes, CH4 resumes execution.

13.7 Overrun Detection Feature

The RTDMA contains overrun detection logic. When the burst for a channel is started, the PERINTFLG is cleared. If however, between the time that the PERINTFLG bit is set by an event trigger and cleared by the start of the burst, an additional event trigger arrives, and the second trigger is lost. This condition sets the OVRFLG bit in the CONTROL register as in Figure 13-7. If the overrun interrupt is enabled, the channel interrupt is generated to the Interrupt Controller module.

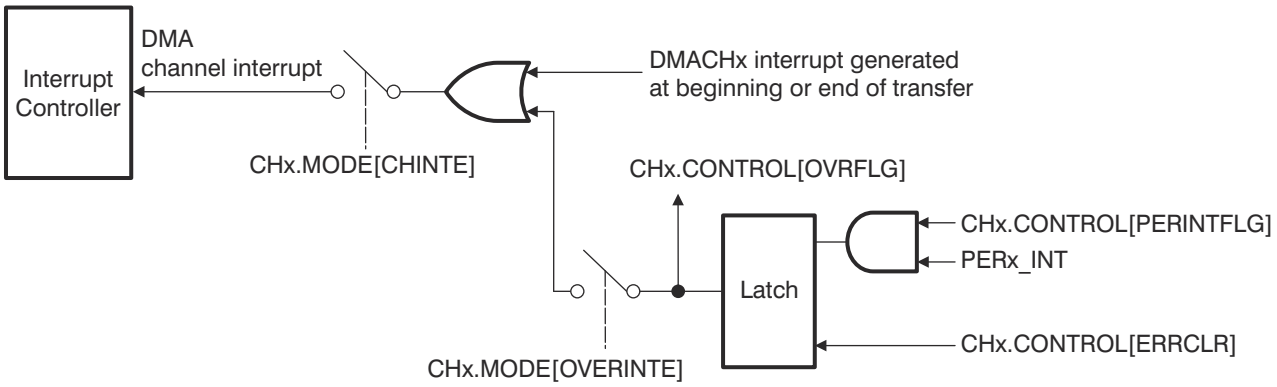


Figure 13-7. Overrun Detection Logic

13.8 Burst Mode

The Real-Time DMA when used with EMIF supports Burst Mode. In this mode, sideband signals are directly connected to the RTDMA RD / WR ports with the bus. These signals identify the bus size and location of the EMIF for which the RTDMA is configured to perform burst writes. By setting this bit as active, the RTDMA is optimized for better performance.

When RTDMA is configured to perform burst writes to the EMIF interface, the transfer size needs to be at least 32-bits. In cases where data size has to be 16-bits in a BURST, the user can first copy the 16-bit data from the source (ex: ADC result) to an SRAM block using a 16-bit access and further move data from the SRAM block to EMIF using 32-bit access . The same concept is applicable where the data size is 8 bits, so the minimum is the transfer size = 4 * DATASIZE.

Note

For best throughput, BURSTSIZE can be configured as a multiple of 4. Burst accesses when BURSTSIZE<4 are treated as normal accesses and not an uninterrupted burst access. Similarly, during an ongoing BURST transaction when BURSTCOUNT reaches < 4 entries, the last transaction is be treated as a normal (non-burst) mode.

Example: If the BURSTSIZE = 10 and DATASIZE = 8 bits, there are 2 burst accesses and the last access with 3 bytes are non-burst access.

Two modes are supported during burst mode operation using the BURST_INTF_CTRL register:

- **Mode 0 (Burst signaling disabled):** Burst Mode is disabled and the RTDMA accesses data as normal.
- **Mode 1 (Reserved)**
- **Mode 2 (Burst signaling enabled):** Burst mode is enabled and cannot be interrupted.

13.9 Safety and Security

The RTDMA has protection mechanisms to provide safety and security for the RTDMA sub-system.

13.9.1 Safety

13.9.1.1 Lockstep Mode

Multiple RTDMA instances can operate in Lockstep to detect faults required to meet safety standards. On a device that supports this configuration, the RTDMA instances are in lockstep by default. In this mode, RTDMA1 individually or RTDMA1/2 in lockstep can be used.CPU3 can configure and service RTDMA independent of CPU1's LPM status (IDLE, STANDBY).

The application can clock gate RTDMA, whether or not the RTDMA is actually operating in lockstep to save power while the CPUx is in lock step configuration and the RTDMA is controlled by the clock enable control (PCLKCR0).

Note

When disabling the default lockstep mode of the device, make sure that the secondary RTDMA instance is in soft reset.

Refer to Chapter 3 and Chapter 5 for information about lockstep settings as well as the device data sheet to see which devices support lockstep.

13.9.1.2 Memory Protection Unit (MPU)

RTDMA has a dedicated, integrated Memory Protection Unit (MPU) which provides memory protection support and prevents illegal accesses. Only the privileged software executing from SROOT (CPU1 LINK2) and CPU1 LINK0/1 can access and define the RTDMA-MPU registers.

The primary function of the MPU is to prevent the RTDMA from accessing regions that are not allocated to the corresponding channels/regions. Key features of the MPU are:

- Associate RTDMA channels to MPU regions.
- Demarcate Read / Write ranges across FLASH, RAM, ROM, and other peripheral memories.
- Detect errors, block access, and trigger an interrupt on any violations.

MPU consists of (16) 4KB regions with individual user-configurable registers to define:

- Start and end addresses
- Read / Write permissions
- Channel Configurations
- Lock / Commit configurations

The region of addresses is set to SROOT only access by the SSU at boot, and these settings cannot be modified by the user afterwards. The start and end addresses (MPUn_RD/WR_START and MPUn_RD/WR_END) are user configurable to any range covering FLASH, RAM, ROM, and the peripherals. MPUn_CHMASK register is used to set association of channels to MPU regions. Each channel can be enabled in one or more MPU regions and each MPU region can be enabled in multiple channels.

Note

MPU_LOCK and MPU_COMMIT are global to all MPU regions and modify access of the MPU_CTRL register. MPUR_LOCK and MPUR_COMMIT are specific to the configured MPU region and modify access of region configuration registers.

In Figure 13-8, the following configuration is depicted for peripheral-to-memory transfer:

- SROOT sets up RTDMA MPU regions to
  - Region 2: Start, End addresses encompass ADCA, ADCB, ADCC - MPU region enabled for Channel 1, Read
  - Region 3: Start, End addresses encompass ADCD, ADCE - MPU region enabled for Channel 2, Read
  - Region 4: Start, End addresses encompass CPU1 Local RAM - MPU region enabled for Channel 1, Write
  - Region 5: Start, End addresses encompass Global RAM - MPU region enabled for Channel 2, Write
- Channel SSU settings
  - RTDMA1 Channel 1: CPU1.APR6 LINK3 Read / Write
  - RTDMA1 Channel 2: CPU2.APR4 LINK1 Read / Write

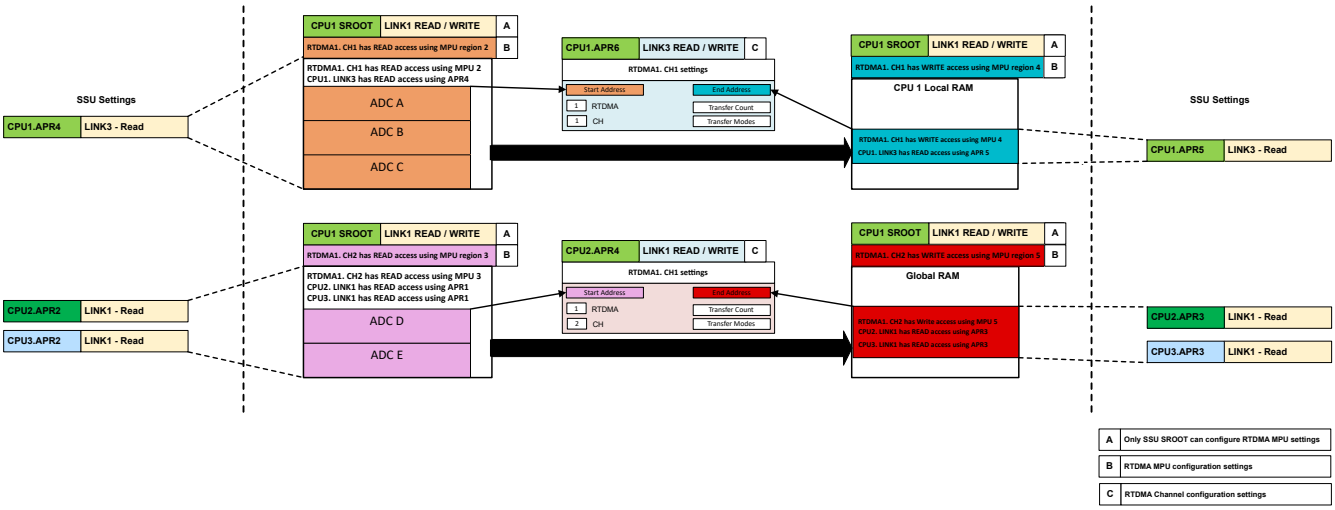


Figure 13-8. Peripheral-to-Memory Transfer Example

In Figure 13-9, the following configuration is depicted for memory-to-memory transfer:

RTDMA1:

- SROOT sets up RTDMA MPU regions to
  - Region 2: Start, End addresses encompass a section of CPU1 Local RAM - MPU region Enabled for RTDMA1.Channel 1 / Channel 4 - Read
  - Region 3: Start, End addresses encompass a section of CPU2 Local RAM - MPU region Enabled for RTDMA1.Channel 1 / Channel 4 - Write
- Channel SSU settings
  - RTDMA1 Channel1: CPU1.APR6.LINK3 - READ / WRITE
  - RTDMA1 Channel4: CPU1.APR6.LINK3 - READ / WRITE

RTDMA2:

- SROOT sets up RTDMA MPU regions to
  - Region 4: Start, End addresses encompass a section of CPU1 Local RAM - MPU region Enabled for RTDMA2.Channel 3 - Write
  - Region 5: Start, End addresses encompass a section of CPU2 Local RAM - MPU region Enabled for RTDMA2.Channel 3 - Read
- Channel SSU settings
  - RTDMA2 Channel3: CPU2.APR4.LINK1 - READ / WRITE

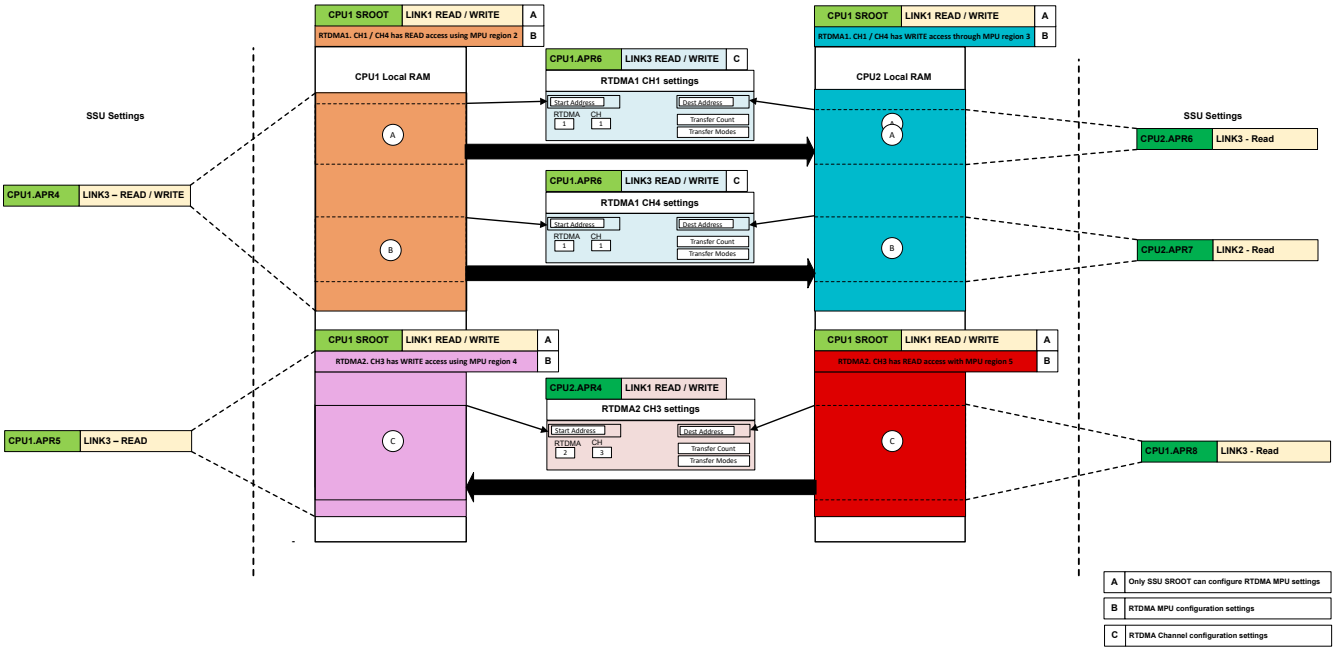


Figure 13-9. Memory-to-Memory Transfer Example

13.9.1.2.1 MPU Errors

MPU monitors all the read and write accesses happening on the bus continuously. Any faults in the interface are communicated to the ESM module where the errors are latched and RTDMA access permissions is determined accordingly.

Errors can be of the following types:

- Address Checks:
  - If an address does not belong to any region, a fault is generated and triggered to the ESM. In this case, access is blocked.
  - More than one MPU regions responded to a data write or read access
    - Multiple regions respond and allow the access – no fault is generated and access goes through.
    - Multiple regions respond and one of the regions blocks the access – fault is generated and triggered to the ESM and access is blocked.
- Access Permission Checks:
  - If an access to a range defined by MPU region does not have the necessary access permissions, a fault is generated and triggered to the ESM and access is blocked.

Note

When SIC and MPU error happens on the same cycle – MPU error address is reported to the ESM. Refer to Section 7.2.3 for more details about how all errors are registered within the ESM.

The MPU is enabled by default. If memory protection support is not needed, disable the MPU during RTDMA initialization.

13.9.2 Security

In addition to MPU specific configurations, the SSU can be used to add another level of security to the RTDMA system, optionally.

RTDMA MPU versus SSU

RTDMA is an independent bus initiator that is not directly governed by SSU access filters. RTDMA receives sideband signals from the SSU that determines LINK ownership of channel configurations. The MPU is used to actually protect data transfers initiated by a DMA channel.

How RTDMA utilizes the SSU

Only the privileged software executing from SROOT (CPU1 LINK2) and CPU1 LINK0/1 can read / write and define the RTDMA MPU registers. Safety and Security Unit (SSU) related checks are placed in a hardcoded region outside of the RTDMA to verify appropriate permissions. User can assign each channel to an AP region inside the SSU to define the channel ownership to a certain link / zone and prevent illegal accesses to channel specific registers from other sources.

Note

MPU register definition and channel association is similar to the SSU architecture.

13.9.3 RTDMA Errors

The RTDMA interfaces are directly connected to the Error Aggregator which aggregates errors occurred during RTDMA data write accesses and RTDMA data read accesses. RTDMA outputs the error, error type, and error address to the Error Aggregator. The high-priority errors from the RD / WR interfaces are combined as RTDMAx HPERR, and the low-priority errors from the RD / WR interfaces are combined as RTDMAx LPERR.

The errors are propagated to the ESM, PIPE, and CPU to take the appropriate further action.

Table 13-3. RTDMAx Error Aggregator Errors

RTDMAx DW Interface Errors	Priority/Description	RTDMAx DR Interface Errors	Priority/Description
Security Violation Error	High Priority - Illegal write to memory or peripherals, Write to an illegal location	Security Violation Error	High Priority - Illegal read of memory or peripherals, Read of an illegal location
Access Timeout Error	High Priority - Mechanism to check for timeout	Access Timeout Error	High Priority - Mechanism to check for timeout
Access ACK Error	High Priority - Any access that does not receive an ACK from the memory controller or peripheral bridge	Access ACK Error	High Priority - Any access that does not receive an ACK from the memory controller or peripheral bridge
		Uncorrectable Error	High Priority - Read accesses that result in a double-bit ECC error.
		Correctable Error	Low Priority - Read accesses that result in a single-bit ECC error.

Note

Refer to Chapter 8 and Chapter 7 for the complete list of error sources and information on how to read and use these errors for debugging purposes.

13.9.4 Self-Test and Diagnostics

RTDMA contains safety diagnostics registers reflecting the architecture of the C29x self-test and diagnostic registers. Parity and ECC logic is implemented for the bus and registers to help achieve overall safety goals.

Self-test logic is used to cover faults local to the ECC module. Once a self-test controller is configured and the CPU initiates diagnostics on the program bus, the controller detects faults in ECC logic and stores these values in the appropriate registers.

Fault emulation logic is used to induce errors and check that the error signal generation is working as expected as a part of diagnostics tests (example, NMI servicing after CPU fault).

Note

Refer to the RTDMA Diagnostics and Self-Test Registers for more information on fault emulation and ECC checker diagnostics capabilities of the RTDMA.

13.10 Software

13.10.1 RTDMA Registers to Driverlib Functions

Table 13-4. RTDMA Registers to Driverlib Functions

File	Driverlib Function
DMACTRL	
rtdma.h	__attribute__
rtdma.h	__attribute__
rtdma.h	__attribute__
DEBUGCTRL	
rtdma.h	__attribute__
REVISION	
-	
SWPRI1	
rtdma.h	__attribute__
SWPRI2	
rtdma.h	__attribute__
PRIORITYSTAT	
-	
DMACFG_LOCK	
rtdma.h	__attribute__
rtdma.h	__attribute__
DMACFG_COMMIT	
rtdma.h	__attribute__
rtdma.h	__attribute__
MPUR_CHMASK(i)	
rtdma.c	voidDMA_configMPURegion
MPUR_START(i)	
rtdma.c	voidDMA_configMPURegion
MPUR_END(i)	
rtdma.c	voidDMA_configMPURegion
MPUR_LOCK(i)	
rtdma.h	__attribute__
rtdma.h	__attribute__

Table 13-4. RTDMA Registers to Driverlib Functions (continued)

File	Driverlib Function
rtdma.h	__attribute__
rtdma.h	__attribute__
MPUR_COMMIT(i)	
rtdma.h	__attribute__
rtdma.h	__attribute__
rtdma.h	__attribute__
MPUR_ACCESS(i)	
rtdma.c	voidDMA_configMPURegion
MPUCTRL	
rtdma.h	__attribute__
rtdma.h	__attribute__
MPUCFG_LOCK	
rtdma.h	__attribute__
rtdma.h	__attribute__
MPUCFG_COMMIT	
rtdma.h	__attribute__
rtdma.h	__attribute__
MODE	
rtdma.c	voidDMA_configMode
rtdma.h	staticinlinevoidDMA_enableTrigger
rtdma.h	staticinlinevoidDMA_disableTrigger
rtdma.h	staticinlinevoidDMA_setInterruptMode
rtdma.h	staticinlinevoidDMA_enableInterrupt
rtdma.h	staticinlinevoidDMA_disableInterrupt
rtdma.h	staticinlinevoidDMA_enableOverrunInterrupt
rtdma.h	staticinlinevoidDMA_disableOverrunInterrupt
CONTROL	
rtdma.h	staticinlinevoidDMA_triggerSoftReset
rtdma.h	staticinlinevoidDMA_forceTrigger
rtdma.h	staticinlinevoidDMA_clearTriggerFlag
rtdma.h	staticinlineboolDMA_getTransferStatusFlag
rtdma.h	staticinlineboolDMA_getBurstStatusFlag
rtdma.h	staticinlineboolDMA_getRunStatusFlag
rtdma.h	staticinlineboolDMA_getOverflowFlag
rtdma.h	staticinlineboolDMA_getTriggerFlagStatus
rtdma.h	staticinlinevoidDMA_startChannel
rtdma.h	staticinlinevoidDMA_stopChannel
rtdma.h	staticinlinevoidDMA_clearErrorFlag
BURST_SIZE	
rtdma.c	voidDMA_configBurst
BURST_COUNT	
-	
SRC_BURST_STEP	
rtdma.c	voidDMA_configBurst
DST_BURST_STEP	

Table 13-4. RTDMA Registers to Driverlib Functions (continued)

File	Driverlib Function
rtdma.c	voidDMA_configBurst
TRANSFER_SIZE	
rtdma.c	voidDMA_configTransfer
TRANSFER_COUNT	
-	
SRC_TRANSFER_STEP	
rtdma.c	voidDMA_configTransfer
DST_TRANSFER_STEP	
rtdma.c	voidDMA_configTransfer
SRC_WRAP_SIZE	
rtdma.c	voidDMA_configWrap
SRC_WRAP_COUNT	
-	
SRC_WRAP_STEP	
rtdma.c	voidDMA_configWrap
DST_WRAP_SIZE	
rtdma.c	voidDMA_configWrap
DST_WRAP_COUNT	
-	
DST_WRAP_STEP	
rtdma.c	voidDMA_configWrap
SRC_BEG_ADDR_SHADOW	
rtdma.c	voidDMA_configAddresses
rtdma.h	staticinlinevoidDMA_configSourceAddress
SRC_ADDR_SHADOW	
rtdma.c	voidDMA_configAddresses
rtdma.h	staticinlinevoidDMA_configSourceAddress
SRC_BEG_ADDR_ACTIVE	
-	
SRC_ADDR_ACTIVE	
-	
DST_BEG_ADDR_SHADOW	
rtdma.c	voidDMA_configAddresses
rtdma.h	staticinlinevoidDMA_configDestAddress
DST_ADDR_SHADOW	
rtdma.c	voidDMA_configAddresses
rtdma.h	staticinlinevoidDMA_configDestAddress
DST_BEG_ADDR_ACTIVE	
-	
DST_ADDR_ACTIVE	
-	
CHSECLAT1	
-	
CHSECLAT2	
-	

Table 13-4. RTDMA Registers to Driverlib Functions (continued)

File	Driverlib Function
BURST_INTF_CTRL	
rtdma.h	staticinlinevoidDMA_setBurstSignalingMode
CHCFG_LOCK	
rtdma.h	staticinlinevoidDMA_lockAllChannelConfig
rtdma.h	staticinlinevoidDMA_unlockAllChannelConfig
rtdma.h	staticinlinevoidDMA_lockChannelConfig
rtdma.h	staticinlinevoidDMA_unlockChannelConfig
CHCFG_COMMIT	
rtdma.h	staticinlinevoidDMA_commitAllChannelConfig
rtdma.h	staticinlinevoidDMA_commitChannelConfig
rtdma.h	__attribute__
FLTEMU_CONFIG	
-	
FLTEMU_ACCGRPSEL	
-	
FLTEMU_BITSEL	
-	
FLTEMU_ADDR	
-	
SELFTEST_DIAG_DATA0	
-	
SELFTEST_DIAG_DATA1	
-	
SELFTEST_DIAG_DATA2	
-	
SELFTEST_DIAG_ECC	
-	
SELFTEST_DIAG_CONTROL	
-	
SELFTEST_DIAG_STATUS	
-	
SELFTEST_DIAG_STATUS_CLR	
-	

13.10.2 RTDMA Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/rtdma

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

13.10.2.1 RTDMA Academy Lab - SINGLE_CORE

FILE: rtdma_academy_lab.c

This example demonstrates how to use the RTDMA to service the ADC in a real-time processing application. The ePWM and ADC modules generate and sample a PWM waveform. The DMA is used to store the ADC samples in a ping pong buffer, so sample groups can be simultaneously processed by the CPU in an ISR.

External Connections

- ADCINA0 (AIO160) to EPWM1_A (GPIO0)

Watch Variables

- *PingPongState* - Ping-pong buffer state.
- *LedCtr* - Counter to slow LED toggling.
- *TaskDelayUs* - Delay to simulate data processing task.
- *OverCnt* - Counter to store DMA overwrites.
- *TimDiff* - To measure the DMA ISR time.
- *AdcBufRaw* - Ping-pong buffer.

13.10.2.2 RTDMA Transfer - SINGLE_CORE

FILE: rtdma_ex1_mem_transfer.c

This example uses one RTDMA channel to transfer data from a buffer to another buffer in RAM. The example triggers the DMA channel repeatedly until the transfer of 16 bursts (where each burst is 4 8-bit words) has been completed. When the whole transfer is complete it will trigger the DMA interrupt. The RTDMA1 MPU is disabled in this example.

Watch Variables

- *TxData* - Data to send
- *RxData* - Received data
- *done* - Transfer successful

13.10.2.3 RTDMA Transfer with MPU - SINGLE_CORE

FILE: rtdma_ex2_mem_transfer_mpu.c

This example uses one RTDMA channel to transfer data from a buffer to another buffer in RAM. The example triggers the DMA channel repeatedly until the transfer of 16 bursts (where each burst is 4 8-bit words) has been completed. When the whole transfer is complete it will trigger the DMA interrupt.

The RTDMA1 MPU is enabled in this example to configure a predefined region that dictates read and write access to the transmit buffer and receive buffer. When a DMA channel attempts to access data at an illegal address, MPU outputs a security violation. Any faults in the interface are communicated to the Error Aggregator module which are sent to the Error Signaling Module (ESM) where the errors are latched and RTDMA access is blocked.

Configured Address Ranges: MPU Region Start Address - 0x200E0000 MPU Region End Address - 0x200E0FFF TxData Start Address - 0x200E0100 TxData End Address - 0x200E013F
RxData Start Address - 0x200E0FDC RxData End Address - 0x200E101B

13.11 RTDMA Registers

This section describes the RTDMA registers.

13.11.1 RTDMA Base Address Table

Table 13-5. RTDMA Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
RTDMA_REGS [1]	RTDMA1_BASE	0x301C_0000	-	YES	YES	YES	-	-	YES	YES
RTDMA_DIAG_REGS [1]	RTDMA1_DIAG_BASE	0x301C_0800	-	YES	YES	YES	-	-	YES	YES
RTDMA_SELFTEST_REGS [1]	RTDMA1_SELFTEST_BASE	0x301C_0C00	-	YES	YES	YES	-	-	YES	YES
RTDMA_MPU_REGS [1]	RTDMA1_MPU_BASE	0x301C_1000	-	YES	YES	YES	-	-	YES	YES
RTDMA_REGS [1]	RTDMA2_BASE	0x301C_8000	-	YES	YES	YES	-	-	YES	YES
RTDMA_DIAG_REGS [1]	RTDMA2_DIAG_BASE	0x301C_8800	-	YES	YES	YES	-	-	YES	YES
RTDMA_SELFTEST_REGS [1]	RTDMA2_SELFTEST_BASE	0x301C_8C00	-	YES	YES	YES	-	-	YES	YES
RTDMA_MPU_REGS [1]	RTDMA2_MPU_BASE	0x301C_9000	-	YES	YES	YES	-	-	YES	YES
RTDMA_CH_REGS	RTDMA1CH1_BASE	0x6000_0000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA1CH2_BASE	0x6000_1000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA1CH3_BASE	0x6000_2000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA1CH4_BASE	0x6000_3000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA1CH5_BASE	0x6000_4000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA1CH6_BASE	0x6000_5000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA1CH7_BASE	0x6000_6000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA1CH8_BASE	0x6000_7000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA1CH9_BASE	0x6000_8000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA1CH10_BASE	0x6000_9000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA2CH1_BASE	0x6001_0000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA2CH2_BASE	0x6001_1000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA2CH3_BASE	0x6001_2000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA2CH4_BASE	0x6001_3000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA2CH5_BASE	0x6001_4000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA2CH6_BASE	0x6001_5000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA2CH7_BASE	0x6001_6000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA2CH8_BASE	0x6001_7000	YES	YES	YES	YES	YES	YES	-	YES

Table 13-5. RTDMA Base Address Table (continued)

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
RTDMA_CH_REGS	RTDMA2CH9_BASE	0x6001_8000	YES	YES	YES	YES	YES	YES	-	YES
RTDMA_CH_REGS	RTDMA2CH10_BASE	0x6001_9000	YES	YES	YES	YES	YES	YES	-	YES

(1) Registers writeable by CPU1.LINK0, CPU1.LINK1, CPU1.LINK2 only. All CPUs can read all registers in all LINKs. Debug write access only allowed if Zone0 or Zone1 are enabled for full debug by all CPUs. Debug reads always allowed. Register Read/Write access by HSM.

13.11.2 RTDMA_REGS Registers

Table 13-6 lists the memory-mapped registers for the RTDMA_REGS registers. All register offset addresses not listed in Table 13-6 should be considered as reserved locations and the register contents should not be modified.

Table 13-6. RTDMA_REGS Registers

Offset	Acronym	Register Name	Protection
0h	DMACTRL	DMA Control Register	LOCK: DMACFG_LOCK.LOCK
4h	DEBUGCTRL	Debug Control Register	LOCK: DMACFG_LOCK.LOCK
8h	REVISION	RTDMA Revision Control Register	
14h	SWPRI1	Software Priority Configuration Register 1	LOCK: DMACFG_LOCK.LOCK
18h	SWPRI2	Software Priority Configuration Register 2	LOCK: DMACFG_LOCK.LOCK
1Ch	PRIORITYSTAT	Priority Status Register	
40h	DMACFG_LOCK	Channel Configuration Temporary Lock	COMMIT: DMACFG_COMMIT.COMMIT
44h	DMACFG_COMMIT	Channel Configuration Permanent Commit	

Complex bit access types are encoded to fit into small table cells. Table 13-7 shows the codes that are used for access types in this section.

Table 13-7. RTDMA_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

13.11.2.1 DMACTRL Register (Offset = 0h) [Reset = 00000000h]

DMACTRL is shown in Figure 13-10 and described in Table 13-8.

Return to the Summary Table.

DMA Control Register

Figure 13-10. DMACTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							PRIORITYSEL
R-0h							R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						PRIORITYRESET	HARDRESET
R-0h						R-0/W1S-0h	R-0/W1S-0h

Table 13-8. DMACTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	Reserved
16	PRIORITYSEL	R/W	0h	The priority select bit: 0: the round-robin priority scheme. 1: Software configurable priority for channels, Priority of the channels is set in the SWPRI reg Reset type: SYSRSn
15-2	RESERVED	R	0h	Reserved
1	PRIORITYRESET	R-0/W1S	0h	If PRIORITYSEL==0: The priority reset bit resets the round-robin state machine when a 1 is written. Service starts from the first enabled channel. Writes of 0 are ignored and this bit always reads back a 0. If PRIORITYSEL==1: The SWPRI register is reset to it's reset value when a 1 is written. All channels will be of priority '1'. Writes of 0 are ignored and this bit always reads back a 0. Reset type: SYSRSn
0	HARDRESET	R-0/W1S	0h	Writing a 1 to the hard reset bit resets the whole DMA and aborts any current access (similar to applying a device reset). Writes of 0 are ignored and this bit always reads back a 0. For a soft reset, a bit is provided for each channel to perform a gentler reset. Refer to the channel control registers. When writing to this bit, there is a one cycle delay before it takes effect. Hence, a one-cycle delay (such as a NOP instruction) is required in software before attempting to access any other DMA register. Reset type: SYSRSn

13.11.2.2 DEBUGCTRL Register (Offset = 4h) [Reset = 00000000h]

DEBUGCTRL is shown in Figure 13-11 and described in Table 13-9.

Return to the Summary Table.

Debug Control Register

Figure 13-11. DEBUGCTRL Register

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

Table 13-9. DEBUGCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	FREE	R/W	0h	Emulation Control This bit specifies the action when an emulation halt event occurs. Reset type: SYSRSn 0h (R/W) = The DMA completes the current read-write operation, then halts. 1h (R/W) = The DMA continues running during an emulation halt.
14-0	RESERVED	R	0h	Reserved

13.11.2.3 REVISION Register (Offset = 8h) [Reset = 00000000h]

REVISION is shown in Figure 13-12 and described in Table 13-10.

Return to the Summary Table.

RTDMA Revision Control Register

Figure 13-12. REVISION Register

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

Table 13-10. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	REV	R	0h	RTDMA Revision. To allow documentation of differences between revisions. First version is labeled as 00h. Reset type: SYSRSn
7-0	TYPE	R	0h	RTDMA Type. Set to 0 for this type Reset type: SYSRSn

13.11.2.4 SWPRI1 Register (Offset = 14h) [Reset = 11111111h]

SWPRI1 is shown in Figure 13-13 and described in Table 13-11.

Return to the Summary Table.

Software Priority Configuration Register 1

Figure 13-13. SWPRI1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH8PRIORITY				CH7PRIORITY				CH6PRIORITY				CH5PRIORITY			
R/W-1h				R/W-1h				R/W-1h				R/W-1h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH4PRIORITY				CH3PRIORITY				CH2PRIORITY				CH1PRIORITY			
R/W-1h				R/W-1h				R/W-1h				R/W-1h			

Table 13-11. SWPRI1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	CH8PRIORITY	R/W	1h	DMA Channel Priority Configuration: Priority can be set to any value of 0x0 to 0x3 Default value is 1 and value of 0 is treated as a special condition during arbitration 0000: Priority 0 0001: Priority 1 0010: Priority 2 0011: Priority 3 0100 to 1111: Reserved Reset type: SYSRSn
27-24	CH7PRIORITY	R/W	1h	DMA Channel Priority Configuration: Priority can be set to any value of 0x0 to 0x3 Default value is 1 and value of 0 is treated as a special condition during arbitration 0000: Priority 0 0001: Priority 1 0010: Priority 2 0011: Priority 3 0100 to 1111: Reserved Reset type: SYSRSn
23-20	CH6PRIORITY	R/W	1h	DMA Channel Priority Configuration: Priority can be set to any value of 0x0 to 0x3 Default value is 1 and value of 0 is treated as a special condition during arbitration 0000: Priority 0 0001: Priority 1 0010: Priority 2 0011: Priority 3 0100 to 1111: Reserved Reset type: SYSRSn
19-16	CH5PRIORITY	R/W	1h	DMA Channel Priority Configuration: Priority can be set to any value of 0x0 to 0x3 Default value is 1 and value of 0 is treated as a special condition during arbitration 0000: Priority 0 0001: Priority 1 0010: Priority 2 0011: Priority 3 0100 to 1111: Reserved Reset type: SYSRSn

Table 13-11. SWPRI1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-12	CH4PRIORITY	R/W	1h	DMA Channel Priority Configuration: Priority can be set to any value of 0x0 to 0x3 Default value is 1 and value of 0 is treated as a special condition during arbitration 0000: Priority 0 0001: Priority 1 0010: Priority 2 0011: Priority 3 0100 to 1111: Reserved Reset type: SYSRSn
11-8	CH3PRIORITY	R/W	1h	DMA Channel Priority Configuration: Priority can be set to any value of 0x0 to 0x3 Default value is 1 and value of 0 is treated as a special condition during arbitration 0000: Priority 0 0001: Priority 1 0010: Priority 2 0011: Priority 3 0100 to 1111: Reserved Reset type: SYSRSn
7-4	CH2PRIORITY	R/W	1h	DMA Channel Priority Configuration: Priority can be set to any value of 0x0 to 0x3 Default value is 1 and value of 0 is treated as a special condition during arbitration 0000: Priority 0 0001: Priority 1 0010: Priority 2 0011: Priority 3 0100 to 1111: Reserved Reset type: SYSRSn
3-0	CH1PRIORITY	R/W	1h	DMA Channel Priority Configuration: Priority can be set to any value of 0x0 to 0x3 Default value is 1 and value of 0 is treated as a special condition during arbitration 0000: Priority 0 0001: Priority 1 0010: Priority 2 0011: Priority 3 0100 to 1111: Reserved Reset type: SYSRSn

13.11.2.5 SWPRI2 Register (Offset = 18h) [Reset = 11111111h]

SWPRI2 is shown in Figure 13-14 and described in Table 13-12.

Return to the Summary Table.

Software Priority Configuration Register 2

Figure 13-14. SWPRI2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				RESERVED				RESERVED				RESERVED			
R/W-1h				R/W-1h				R/W-1h				R/W-1h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				RESERVED				CH10PRIORITY				CH9PRIORITY			
R/W-1h				R/W-1h				R/W-1h				R/W-1h			

Table 13-12. SWPRI2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	1h	Reserved
27-24	RESERVED	R/W	1h	Reserved
23-20	RESERVED	R/W	1h	Reserved
19-16	RESERVED	R/W	1h	Reserved
15-12	RESERVED	R/W	1h	Reserved
11-8	RESERVED	R/W	1h	Reserved
7-4	CH10PRIORITY	R/W	1h	DMA Channel Priority Configuration: Priority can be set to any value of 0x0 to 0x3 Default value is 1 and value of 0 is treated as a special condition during arbitration 0000: Priority 0 0001: Priority 1 0010: Priority 2 0011: Priority 3 0100 to 1111: Reserved Reset type: SYSRSn
3-0	CH9PRIORITY	R/W	1h	DMA Channel Priority Configuration: Priority can be set to any value of 0x0 to 0x3 Default value is 1 and value of 0 is treated as a special condition during arbitration 0000: Priority 0 0001: Priority 1 0010: Priority 2 0011: Priority 3 0100 to 1111: Reserved Reset type: SYSRSn

13.11.2.6 PRIORITYSTAT Register (Offset = 1Ch) [Reset = 00000000h]

PRIORITYSTAT is shown in Figure 13-15 and described in Table 13-13.

Return to the Summary Table.

Priority Status Register

Figure 13-15. PRIORITYSTAT Register

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

Table 13-13. PRIORITYSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	Reserved
12-8	ACTIVESTS_SHADOW	R	0h	Active Channel Status Shadow These bits are only meaningful when Channel in high-priority mode (SWPRIx[CHyPRIORITY]==0). When such high priority channel is serviced, the ACTIVESTS bits are copied to the shadow bits and indicate which channel was interrupted by high priority channel. When high priority channel service is completed, the shadow bits are copied back to the ACTIVESTS bits. If this bit field is zero or the same as the ACTIVESTS bit field, then no channel is pending due to a high priority Channel interrupt. 00000: No channel is active 00001: CH1 is active 00010: CH2 is active ... ... 01010: CH10 is active 01011: Reserved ... ... 11111: Reserved Reset type: SYSRSn
7-5	RESERVED	R	0h	Reserved

Table 13-13. PRIORITYSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	ACTIVESTS	R	0h	Active Channel Status These bits indicate which channel (if any) is currently active or performing a transfer. 00000: No channel is active 00001: CH1 is active 00010: CH2 is active ... ... 01010: CH10 is active 01011: Reserved ... ... 11111: Reserved Reset type: SYSRSn

13.11.2.7 DMACFG_LOCK Register (Offset = 40h) [Reset = 00000000h]

DMACFG_LOCK is shown in Figure 13-16 and described in Table 13-14.

Return to the Summary Table.

Channel Configuration Temporary Lock

Figure 13-16. DMACFG_LOCK Register

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

Table 13-14. DMACFG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks configuration registers in the RTDMA_REGS aperture DMACTL,DEBUGCTL, SWPRI1/2 (writes will have no effect on them). This bit can only be modified if DMACFG_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: SYSRSn

13.11.2.8 DMACFG_COMMIT Register (Offset = 44h) [Reset = 00000000h]

DMACFG_COMMIT is shown in Figure 13-17 and described in Table 13-15.

Return to the Summary Table.

Channel Configuration Permanent Commit

Figure 13-17. DMACFG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 13-15. DMACFG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the DMACFG_LOCK register. This bit cannot be cleared, except by reset. 0 : DMACFG_LOCK is modifiable 1 : DMACFG_LOCK is committed permanently Reset type: SYSRSn

13.11.3 RTDMA_DIAG_REGS Registers

Table 13-16 lists the memory-mapped registers for the RTDMA_DIAG_REGS registers. All register offset addresses not listed in Table 13-16 should be considered as reserved locations and the register contents should not be modified.

Table 13-16. RTDMA_DIAG_REGS Registers

Offset	Acronym	Register Name	Protection
0h	FLTEMU_CONFIG	Fault emulation configuration registerr	KEY:KEY=0xa5
4h	FLTEMU_ACCGRPSEL	Fault emulation access information group selection register	
8h	FLTEMU_BITSEL	Fault emulation bitsel	
Ch	FLTEMU_ADDR	Fault emulation access address register	

Complex bit access types are encoded to fit into small table cells. Table 13-17 shows the codes that are used for access types in this section.

Table 13-17. RTDMA_DIAG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

13.11.3.1 FLTEMU_CONFIG Register (Offset = 0h) [Reset = 00000000h]

FLTEMU_CONFIG is shown in Figure 13-18 and described in Table 13-18.

Return to the Summary Table.

Fault emulation configuration register

Figure 13-18. FLTEMU_CONFIG Register

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

Table 13-18. FLTEMU_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	KEY	W	0h	Write Key In order to write to any bit in this register, 0xA5 must be written to this field at the same time. Otherwise, writes are ignored. The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
7-2	RESERVED	R	0h	Reserved
1	DBL_BIT_INJ_EN	R/W	0h	0' : Fault emulation on data buses (chosen using [FLTEMU_ACCGRPSEL]DATA_GROUP_SEL) happens on the data bit field = [FLTEMU]BITSEL '1' : Fault emulation on data buses (chosen using [FLTEMU_ACCGRPSEL]DATA_GROUP_SEL) happens on the data bit fields [FLTEMU]BITSEL and [FLTEMU]BITSEL + 1 Note: It should be ensured that both data bits where fault injection happens belong to same data checker. For example, if [FLTEMU]BITSEL = 31 and a 32 bit check is being done on the data, then uncorrectable error is not generated as bit31 and bit32 are not checked in the same 32b checker and so a correctable error is generated instead Reset type: SYSRSn
0	ENABLE	R/W	0h	1' : Fault emulation enable '0' : Fault emulation disable. When this bit is '0', the fault injection is disabled. To enable error indication when this bit is '1', SIC_CONFIG.E2E_EN has to be set to '1' Reset type: SYSRSn

13.11.3.2 FLTEMU_ACCGRPSEL Register (Offset = 4h) [Reset = 00000000h]

FLTEMU_ACCGRPSEL is shown in Figure 13-19 and described in Table 13-19.

Return to the Summary Table.

Fault emulation access information group selection register

Figure 13-19. FLTEMU_ACCGRPSEL Register

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

Table 13-19. FLTEMU_ACCGRPSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	DATA_GROUP_SEL	R/W	0h	00 : Reserved 01: DMA Read interface 02: DMA Write interface All other values are reserved Note: To avoid spurious error injections, CTRL_GROUP_SEL field is recommended to be kept zero for Data group fault emulation. Reset type: SYSRSn
7-0	CTRL_GROUP_SEL	R/W	0h	00 : Reserved 01: DMA Read interface 02: DMA Write interface All other values are reserved Note: To avoid spurious error injections, DATA_GROUP_SEL field is recommended to be kept zero for CTRL group fault emulation. Reset type: SYSRSn

13.11.3.3 FLTEMU_BITSEL Register (Offset = 8h) [Reset = 00000000h]

FLTEMU_BITSEL is shown in Figure 13-20 and described in Table 13-20.

Return to the Summary Table.

Fault emulation bitsel

Figure 13-20. FLTEMU_BITSEL Register

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

Table 13-20. FLTEMU_BITSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6-0	BITSEL	R/W	0h	For Control group, this field is interpreted as follows: 0: Fault injection happens on ACK received from chip resource 1 : Fault injection happens on ADDR bit 0 2 : Fault injection happens on ADDR bit 1 .. 32 : Fault injection happens on ADDR bit 31 33: Fault injection happens on BYTEEN bit 0 .. 40 : Fault injection happens on BYTEEN bit 7 41 : Fault injection happens on BYTEEN bit 8 (only applicable for program fetch access) .. 48 : Fault injection happens on BYTEEN bit 15 (only applicable for program fetch access) 49 : Fault injection happens on SIZE bit 0 50 : Fault injection happens on SIZE bit 1 .. 53 : Fault injection happens on SIZE bit 4 54 : Fault injection happens on SIZE bit 5 (only applicable for program fetch access) 55: Fault injection happens on READY received from chip resource ---------- For data group, this field is interpreted as follows: 0: Fault injection happens on data bit 0 1: Fault injection happens on data bit 1 2: Fault injection happens on data bit 2 .. 63: Fault injection happens on data bit 63 64: Fault injection happens on data bit 64 (applicable only for fetch access) 65: Fault injection happens on data bit 65 (applicable only for fetch access) .. 127: Fault injection happens on data bit 127 (applicable only for fetch access) Reset type: SYSRSn

13.11.3.4 FLTEMU_ADDR Register (Offset = Ch) [Reset = 00000000h]

FLTEMU_ADDR is shown in Figure 13-21 and described in Table 13-21.

Return to the Summary Table.

Fault emulation access address register

Figure 13-21. FLTEMU_ADDR Register

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

Table 13-21. FLTEMU_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R/W	0h	Fault emulation is performed for accesses with this address Reset type: SYSRSn

13.11.4 RTDMA_SELFTEST_REGS Registers

Table 13-22 lists the memory-mapped registers for the RTDMA_SELFTEST_REGS registers. All register offset addresses not listed in Table 13-22 should be considered as reserved locations and the register contents should not be modified.

Table 13-22. RTDMA_SELFTEST_REGS Registers

Offset	Acronym	Register Name	Protection
0h	SELFTEST_DIAG_DATA0	Diagnostics data register 0	
4h	SELFTEST_DIAG_DATA1	Diagnostics data register 1	
8h	SELFTEST_DIAG_DATA2	Diagnostics data register 2	
20h	SELFTEST_DIAG_ECC	Diagnostics ECC	
28h	SELFTEST_DIAG_CONTROL	Diagnostic test enable	
2Ch	SELFTEST_DIAG_STATUS	Diagnostic status register	
30h	SELFTEST_DIAG_STATUS_CLR	Diagnostic status clear register	

Complex bit access types are encoded to fit into small table cells. Table 13-23 shows the codes that are used for access types in this section.

Table 13-23. RTDMA_SELFTEST_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

13.11.4.1 SELFTEST_DIAG_DATA0 Register (Offset = 0h) [Reset = 00000000h]

SELFTEST_DIAG_DATA0 is shown in Figure 13-22 and described in Table 13-24.

Return to the Summary Table.

Diagnostics data register 0

Figure 13-22. SELFTEST_DIAG_DATA0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SELFTEST_DIAG_DATA0																															
R/W-0h																															

Table 13-24. SELFTEST_DIAG_DATA0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SELFTEST_DIAG_DATA0	R/W	0h	Self test Diagnostics data 0 This register is used to specify the [31:0] bits of the data to perform Self test ECC checker diagnostics. Reset type: SYSRSn

13.11.4.2 SELFTEST_DIAG_DATA1 Register (Offset = 4h) [Reset = 00000000h]

SELFTEST_DIAG_DATA1 is shown in Figure 13-23 and described in Table 13-25.

Return to the Summary Table.

Diagnostics data register 1

Figure 13-23. SELFTEST_DIAG_DATA1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SELFTEST_DIAG_DATA1																															
R/W-0h																															

Table 13-25. SELFTEST_DIAG_DATA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SELFTEST_DIAG_DATA1	R/W	0h	Self test Diagnostics data 1 This register is used to specify the [63:32] bits of the data to perform Self test ECC checker diagnostics. Reset type: SYSRSn

13.11.4.3 SELFTEST_DIAG_DATA2 Register (Offset = 8h) [Reset = 00000000h]

SELFTEST_DIAG_DATA2 is shown in Figure 13-24 and described in Table 13-26.

Return to the Summary Table.

Diagnostics data register 1

Figure 13-24. SELFTEST_DIAG_DATA2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															SELFTEST_DIAG_DATA2
R-0h															R/W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SELFTEST_DIAG_DATA2															
R/W-0h															

Table 13-26. SELFTEST_DIAG_DATA2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	Reserved
16-0	SELFTEST_DIAG_DATA2	R/W	0h	Self test Diagnostics data 2 This register is used to specify the [81:64] bits of the data to perform Self test ECC checker diagnostics. Reset type: SYSRSn

13.11.4.4 SELFTEST_DIAG_ECC Register (Offset = 20h) [Reset = 00000000h]

SELFTEST_DIAG_ECC is shown in Figure 13-25 and described in Table 13-27.

Return to the Summary Table.

Diagnostics ECC

Figure 13-25. SELFTEST_DIAG_ECC Register

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

Table 13-27. SELFTEST_DIAG_ECC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	SELFTEST_DIAG_ECC	R/W	0h	Self test Diagnostics ECC This register is used to specify the ECC to perform Self test ECC checker diagnostics Reset type: SYSRSn

13.11.4.5 SELFTEST_DIAG_CONTROL Register (Offset = 28h) [Reset = 51080000h]

SELFTEST_DIAG_CONTROL is shown in Figure 13-26 and described in Table 13-28.

Return to the Summary Table.

Enable diagnostic test

Figure 13-26. SELFTEST_DIAG_CONTROL Register

31	30	29	28	27	26	25	24
DIAG_DATA_WIDTH							
R/W-51h							
23	22	21	20	19	18	17	16
RESERVED				DIAG_ECC_WIDTH			
R-0h				R/W-8h			
15	14	13	12	11	10	9	8
RESERVED						DIAG_CHECKER_SEL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
RESERVED	DIAG_SAFETY_SEL	RESERVED		DIAG_TEST_EN			
R-0h	R/W-0h	R-0h		R/W-0h			

Table 13-28. SELFTEST_DIAG_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DIAG_DATA_WIDTH	R/W	51h	ECC data width - Maximum configurable width is 81(0x51) Reset type: SYSRSn
23-20	RESERVED	R	0h	Reserved
19-16	DIAG_ECC_WIDTH	R/W	8h	ECC bit width - Maximum configurable width is 8 (0x8) Reset type: SYSRSn
15-10	RESERVED	R	0h	Reserved
9-8	DIAG_CHECKER_SEL	R/W	0h	This field is used to select the ECC checker. 00 : 16-bit ECC check 01 : 32-bit ECC check 10 : 64-bit ECC check Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6	DIAG_SAFETY_SEL	R/W	0h	Diagnostic safety selection 0 - ECC 1 - Parity Reset type: SYSRSn
5-4	RESERVED	R	0h	Reserved
3-0	DIAG_TEST_EN	R/W	0h	Enable self test mechanism 0011 : Enable self test This field will be '0000' once test done. User needs to write into this register again for next test. Note: Self test should not to be started in the middle of a transfer. Reset type: SYSRSn

13.11.4.6 SELFTEST_DIAG_STATUS Register (Offset = 2Ch) [Reset = 00000000h]

SELFTEST_DIAG_STATUS is shown in Figure 13-27 and described in Table 13-29.

Return to the Summary Table.

Diagnostic status

Figure 13-27. SELFTEST_DIAG_STATUS Register

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

Table 13-29. SELFTEST_DIAG_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	DIAG_FAIL_BIT_INDEX	R	0h	This field is used to specify the position of the flipped bit when test failed. For 2 bit flips, this field points the bit position of the first bit. The second bit will be always adjacent to the first bit. This field will clear when next test configured. Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6-5	DIAG_FAIL_CHECK_TYPE	R	0h	00 : Positive check 01 : Flips one bit 10 : Flips two bit 11 : Reserved Reset type: SYSRSn
4	DIAG_FAIL_UC_ERROR	R	0h	This field is used to specify the diagnostic uncorrectable error when Test failed. Reset type: SYSRSn
3	DIAG_FAIL_C_ERROR	R	0h	This field is used to specify the diagnostic correctable error when Test failed. Reset type: SYSRSn
2	DIAG_TEST_FAIL	R	0h	1 : Test failed (Unexpected error events(C_ERROR/UC_ERROR) occurred during self test) 0 : Test passed Reset type: SYSRSn
1	DIAG_TEST_DONE	R	0h	Completed self test. Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

13.11.4.7 SELFTEST_DIAG_STATUS_CLR Register (Offset = 30h) [Reset = 00000000h]

SELFTEST_DIAG_STATUS_CLR is shown in Figure 13-28 and described in Table 13-30.

Return to the Summary Table.

Diagnostic status clear

Figure 13-28. SELFTEST_DIAG_STATUS_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					DIAG_TEST_FAIL	DIAG_TEST_DONE	RESERVED
R-0h					R-0/W1S-0h	R-0/W1S-0h	R-0h

Table 13-30. SELFTEST_DIAG_STATUS_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	DIAG_TEST_FAIL	R-0/W1S	0h	Clear Test failed status flags 0: Writing a 0 has no effect. 1: Writing a 1 will clear the fields SELFTEST_DIAG_STATUS[DIAG_TEST_FAIL], SELFTEST_DIAG_STATUS[DIAG_FAIL_C_ERROR], SELFTEST_DIAG_STATUS[DIAG_FAIL_UC_ERROR], SELFTEST_DIAG_STATUS[DIAG_FAIL_CHECK_TYPE]. Reset type: SYSRSn
1	DIAG_TEST_DONE	R-0/W1S	0h	Clear self test done status flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the SELFTEST_DIAG_STATUS[DIAG_TEST_DONE] bit. Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

13.11.5 RTDMA_MPU_REGS Registers

Table 13-31 lists the memory-mapped registers for the RTDMA_MPU_REGS registers. All register offset addresses not listed in Table 13-31 should be considered as reserved locations and the register contents should not be modified.

Table 13-31. RTDMA_MPU_REGS Registers

Offset	Acronym	Register Name	Protection
0h + formula	MPUR_CHMASK	MPU Region Configuration	LOCK: MPUR_LOCK.LOCK
4h + formula	MPUR_START_j	MPU Region Start Address	LOCK: MPUR_LOCK.LOCK
8h + formula	MPUR_END_j	MPU Region End Address	LOCK: MPUR_LOCK.LOCK
Ch + formula	MPUR_LOCK_j	MPU Temporary Lock	COMMIT: MPUR_COMMIT.COMMIT
10h + formula	MPUR_COMMIT_j	MPU Permanent Commit	
14h + formula	MPUR_ACCESS_j	MPU Region R/W Access Permissions	LOCK: MPUR_LOCK.LOCK
800h	MPUCTRL	MPU Control Register	LOCK: MPUCFG_LOCK.LOCK
820h	MPUCFG_LOCK	Channel Configuration Temporary Lock	COMMIT: MPUCFG_COMMIT.COMMIT
824h	MPUCFG_COMMIT	Channel Configuration Permanent Commit	

Complex bit access types are encoded to fit into small table cells. Table 13-32 shows the codes that are used for access types in this section.

Table 13-32. RTDMA_MPU_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

13.11.5.1 MPUR_CHMASK Register (Offset = 0h + formula) [Reset = 00000000h]

MPUR_CHMASK is shown in Figure 13-29 and described in Table 13-33.

Return to the Summary Table.

MPU Region Configuration

Offset = 0h + (j * 20h); where j = 0h to Fh

Figure 13-29. MPUR_CHMASK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CH10MASK	CH9MASK
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CH8MASK	CH7MASK	CH6MASK	CH5MASK	CH4MASK	CH3MASK	CH2MASK	CH1MASK
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 13-33. MPUR_CHMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	CH10MASK	R/W	0h	0: Channel access disabled for the region 1: Channel access enabled for the region Reset type: SYSRSn
8	CH9MASK	R/W	0h	0: Channel access disabled for the region 1: Channel access enabled for the region Reset type: SYSRSn
7	CH8MASK	R/W	0h	0: Channel access disabled for the region 1: Channel access enabled for the region Reset type: SYSRSn
6	CH7MASK	R/W	0h	0: Channel access disabled for the region 1: Channel access enabled for the region Reset type: SYSRSn
5	CH6MASK	R/W	0h	0: Channel access disabled for the region 1: Channel access enabled for the region Reset type: SYSRSn
4	CH5MASK	R/W	0h	0: Channel access disabled for the region 1: Channel access enabled for the region Reset type: SYSRSn

Table 13-33. MPUR_CHMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	CH4MASK	R/W	0h	0: Channel access disabled for the region 1: Channel access enabled for the region Reset type: SYSRSn
2	CH3MASK	R/W	0h	0: Channel access disabled for the region 1: Channel access enabled for the region Reset type: SYSRSn
1	CH2MASK	R/W	0h	0: Channel access disabled for the region 1: Channel access enabled for the region Reset type: SYSRSn
0	CH1MASK	R/W	0h	0: Channel access disabled for the region 1: Channel access enabled for the region Reset type: SYSRSn

13.11.5.2 MPUR_START_j Register (Offset = 4h + formula) [Reset = 00000000h]

MPUR_START_j is shown in Figure 13-30 and described in Table 13-34.

Return to the Summary Table.

MPU Region Start Address

Offset = 4h + (j * 20h); where j = 0h to Fh

Figure 13-30. MPUR_START_j Register

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

Table 13-34. MPUR_START_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ADDRH	R/W	0h	Upper 16-bit of the start address, DMA will have access to peripherals, SRAMs, FLASH and external memories. Reset type: SYSRSn
15-12	ADDRL	R/W	0h	Address range start address (granularity of 4KB) within the memory type specified. The values available depends on the memory type chosen and the memory footprint of the device. Reset type: SYSRSn
11-0	RESERVED	R	0h	Reserved

13.11.5.3 MPUR_END_j Register (Offset = 8h + formula) [Reset = 00000000h]

MPUR_END_j is shown in Figure 13-31 and described in Table 13-35.

Return to the Summary Table.

MPU Region End Address

Offset = 8h + (j * 20h); where j = 0h to Fh

Figure 13-31. MPUR_END_j Register

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

Table 13-35. MPUR_END_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ADDRH	R/W	0h	Upper 16-bit of the end address, DMA will have access to peripherals, SRAMs, FLASH and external memories. Reset type: SYSRSn
15-12	ADDRL	R/W	0h	Address range start address (granularity of 4KB) within the memory type specified. The values available depends on the memory type chosen and the memory footprint of the device. Note: The 11:0 bits of the MPU will read as 0x000, but internally are treated as 0xFFF to enable a minimum 4KB boundary for every MPU region. Reset type: SYSRSn
11-0	RESERVED	R	0h	Reserved

13.11.5.4 MPUR_LOCK_j Register (Offset = Ch + formula) [Reset = 00000000h]

MPUR_LOCK_j is shown in Figure 13-32 and described in Table 13-36.

Return to the Summary Table.

MPU Temporary Lock

Offset = Ch + (j * 20h); where j = 0h to Fh

Figure 13-32. MPUR_LOCK_j Register

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

Table 13-36. MPUR_LOCK_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks this MPU registers (writes will have no effect on them). This bit can only be modified if MPU_COMMIT.COMMIT is cleared. simultaneously. 0 : Unlocked 1 : Locked Reset type: SYSRSn

13.11.5.5 MPUR_COMMIT_j Register (Offset = 10h + formula) [Reset = 00000000h]

MPUR_COMMIT_j is shown in Figure 13-33 and described in Table 13-37.

Return to the Summary Table.

MPU Permanent Commit

Offset = 10h + (j * 20h); where j = 0h to Fh

Figure 13-33. MPUR_COMMIT_j Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 13-37. MPUR_COMMIT_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the MPU_LOCK register. This bit cannot be cleared, except by reset. 0 : MPU_LOCK is modifiable 1 : MPU_LOCK is committed permanently Reset type: SYSRSn

13.11.5.6 MPUR_ACCESS_j Register (Offset = 14h + formula) [Reset = 00000000h]

MPUR_ACCESS_j is shown in Figure 13-34 and described in Table 13-38.

Return to the Summary Table.

MPU Region R/W Access Permissions

Offset = 14h + (j * 20h); where j = 0h to Fh

Figure 13-34. MPUR_ACCESS_j Register

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

Table 13-38. MPUR_ACCESS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	ACCESS	R/W	0h	Enables the type of accesses allowed in this region. 00 : No access 01 : Read Access 10 : Read/Write Access 11 : Read/Write Access Reset type: SYSRSn

13.11.5.7 MPUCTRL Register (Offset = 800h) [Reset = 00000001h]

MPUCTRL is shown in Figure 13-35 and described in Table 13-39.

Return to the Summary Table.

MPU Control Register

Figure 13-35. MPUCTRL Register

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

Table 13-39. MPUCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	MPUEN	R/W	1h	This register can only be modified by SROOT. 0 : MPU function disabled 1 : MPU function Enabled Reset type: SYSRSn

13.11.5.8 MPUCFG_LOCK Register (Offset = 820h) [Reset = 00000000h]

MPUCFG_LOCK is shown in Figure 13-36 and described in Table 13-40.

Return to the Summary Table.

Channel Configuration Temporary Lock

Figure 13-36. MPUCFG_LOCK Register

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

Table 13-40. MPUCFG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks this corresponding MPU configuration register MPUCTRL (writes will have no effect on them). This bit can only be modified if MPUCFG_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: SYSRSn

13.11.5.9 MPUCFG_COMMIT Register (Offset = 824h) [Reset = 00000000h]

MPUCFG_COMMIT is shown in Figure 13-37 and described in Table 13-41.

Return to the Summary Table.

Channel Configuration Permanent Commit

Figure 13-37. MPUCFG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 13-41. MPUCFG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the MPUCFG_LOCK register. This bit cannot be cleared, except by reset. 0 : MPUCFG_LOCK is modifiable 1 : MPUCFG_LOCK is committed permanently Reset type: SYSRSn

13.11.6 RTDMA_CH_REGS Registers

Table 13-42 lists the memory-mapped registers for the RTDMA_CH_REGS registers. All register offset addresses not listed in Table 13-42 should be considered as reserved locations and the register contents should not be modified.

Table 13-42. RTDMA_CH_REGS Registers

Offset	Acronym	Register Name	Protection
0h	MODE	Mode Register	LOCK: CHCFG_LOCK.LOCK
4h	CONTROL	Control Register	
8h	BURST_SIZE	Burst Size Register	LOCK: CHCFG_LOCK.LOCK
Ch	BURST_COUNT	Burst Count Register	
10h	SRC_BURST_STEP	Source Burst Step Register	LOCK: CHCFG_LOCK.LOCK
14h	DST_BURST_STEP	Destination Burst Step Register	LOCK: CHCFG_LOCK.LOCK
18h	TRANSFER_SIZE	Transfer Size Register	LOCK: CHCFG_LOCK.LOCK
1Ch	TRANSFER_COUNT	Transfer Count Register	
20h	SRC_TRANSFER_STEP	Source Transfer Step Register	LOCK: CHCFG_LOCK.LOCK
24h	DST_TRANSFER_STEP	Destination Transfer Step Register	LOCK: CHCFG_LOCK.LOCK
28h	SRC_WRAP_SIZE	Source Wrap Size Register	LOCK: CHCFG_LOCK.LOCK
2Ch	SRC_WRAP_COUNT	Source Wrap Count Register	
30h	SRC_WRAP_STEP	Source Wrap Step Register	LOCK: CHCFG_LOCK.LOCK
34h	DST_WRAP_SIZE	Destination Wrap Size Register	LOCK: CHCFG_LOCK.LOCK
38h	DST_WRAP_COUNT	Destination Wrap Count Register	
3Ch	DST_WRAP_STEP	Destination Wrap Step Register	LOCK: CHCFG_LOCK.LOCK
40h	SRC_BEG_ADDR_SHADOW	Source Begin Address Shadow Register	LOCK: CHCFG_LOCK.LOCK
44h	SRC_ADDR_SHADOW	Source Address Shadow Register	LOCK: CHCFG_LOCK.LOCK
48h	SRC_BEG_ADDR_ACTIVE	Source Begin Address Active Register	
4Ch	SRC_ADDR_ACTIVE	Source Address Active Register	
50h	DST_BEG_ADDR_SHADOW	Destination Begin Address Shadow Register	LOCK: CHCFG_LOCK.LOCK

Table 13-42. RTDMA_CH_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
54h	DST_ADDR_SHADOW	Destination Address Shadow Register	LOCK: CHCFG_LOCK.LOCK
58h	DST_BEG_ADDR_ACTIVE	Destination Begin Address Active Register	
5Ch	DST_ADDR_ACTIVE	Destination Address Active Register	
80h	CHSECLAT1	Channel Security Details Latch Register	
84h	CHSECLAT2	Channel Security Details Latch Register	
A0h	BURST_INTF_CTRL	Burst Interface Control Register	LOCK: CHCFG_LOCK.LOCK
100h	CHCFG_LOCK	Channel Configuration Temporary Lock	COMMIT: CHCFG_COMMIT.COMMIT
104h	CHCFG_COMMIT	Channel Configuration Permanent Commit	

Complex bit access types are encoded to fit into small table cells. Table 13-43 shows the codes that are used for access types in this section.

Table 13-43. RTDMA_CH_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

13.11.6.1 MODE Register (Offset = 0h) [Reset = 00000000h]

MODE is shown in Figure 13-38 and described in Table 13-44.

Return to the Summary Table.

Mode Register

Figure 13-38. MODE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED		WRT_DATASIZE		CHINTE	DATASIZE		RESERVED
R-0h		R/W-0h		R/W-0h	R/W-0h		R/W-0h
15	14	13	12	11	10	9	8
RESERVED	CONTINUOUS	ONESHOT	CHINTMODE	PERINTE	OVRINTE	RESERVED	
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	
7	6	5	4	3	2	1	0
PERINTSEL							
R/W-0h							

Table 13-44. MODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	Reserved
21-20	WRT_DATASIZE	R/W	0h	Write Data Size Mode Bit This bit determines whether the DMA channel transfers on the Data Write port are 8 bits, 16 bits, 32 bits or 64 of data per read/write operation. Regardless of this setting, all data lengths and offsets in other DMA registers refer to 8- bit words(Bytes). The pointer step increments must be configured to accommodate 8, 16, 32, 64-bit words. 00: DATASIZE_8 01: DATASIZE_16 10: DATASIZE_32 11: DATASIZE_64 DATASIZE >= WR_DATASIZE. i.e. DST size to be integral multiple of SRC. If DATASIZE_64 then, WR_DATASIZE should be configured to WR_DATASIZE_64 or WR_DATASIZE_32 or WR_DATASIZE_16 or WR_DATASIZE_8 If DATASIZE_32 then, WR_DATASIZE should be configured to WR_DATASIZE_32 or WR_DATASIZE_16 or WR_DATASIZE_8 if DATA_SIZE_16 then, WR_DATASIZE should be configured to WR_DATASIZE_16 or WR_DATASIZE_8 If DATA_SIZE_8 then WR_DATASIZE should be configured to WR_DATASIZE_8 Reset type: SYSRSn
19	CHINTE	R/W	0h	Channel Interrupt Enable Bit This bit enables the DMA channel's CPU interrupt. Reset type: SYSRSn 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled

Table 13-44. MODE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18-17	DATASIZE	R/W	0h	Data Size Mode Bit This bit determines whether the DMA channel transfers 8 bits, 16 bits, 32 bits or 64 of data per read/write operation. Regardless of this setting, all data lengths and offsets in other DMA registers refer to 8-bit words (Bytes). The pointer step increments must be configured to accommodate 8, 16, 32, 64-bit words. 00: DATASIZE_8 01: DATASIZE_16 10: DATASIZE_32 11: DATASIZE_64 Reset type: SYSRSn
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	CONTINUOUS	R/W	0h	Continuous Mode Bit If this bit is set to 1, then the channel re-initializes when TRANSFER_COUNT is zero and waits for the next event trigger. Otherwise, the DMA stops and clears the RUNSTS bit. Reset type: SYSRSn
13	ONESHOT	R/W	0h	One Shot Mode If this bit is set to 1, each peripheral event trigger causes the channel to perform an entire transfer. Otherwise, the channel only performs one burst per trigger. Reset type: SYSRSn
12	CHINTMODE	R/W	0h	Channel Interrupt Generation Mode This bit specifies when the DMA channel generates a CPU interrupt for a transfer. Reset type: SYSRSn 0h (R/W) = Generate interrupt at beginning of new transfer 1h (R/W) = Generate interrupt at end of transfer.
11	PERINTE	R/W	0h	Peripheral Event Trigger Enable This bit enables peripheral event triggers on the DMA channel. Reset type: SYSRSn 0h (R/W) = Peripheral event trigger disabled. Neither the selected peripheral nor software can start a DMA burst. 1h (R/W) = Peripheral event trigger enabled.
10	OVRINTE	R/W	0h	Overflow Interrupt Enable The bit determines whether the DMA module generates a CPU interrupt when it detects an overflow event. Reset type: SYSRSn 0h (R/W) = Overflow interrupt disabled 1h (R/W) = Overflow interrupt enabled
9-8	RESERVED	R	0h	Reserved
7-0	PERINTSEL	R/W	0h	Peripheral Event Trigger Source Select Selects the Trigger and Sync Source of the DMA Channel 0: No Peripheral Connection 1..239: Details in DMA trigger select mux of the device 240: CH1 Interrupt (Reserved for CH1) 241: CH2 Interrupt (Reserved for CH2) 242: CH3 Interrupt (Reserved for CH3) ... ... 249: CH10 Interrupt (Reserved for CH10) 250: Reserved ... ... 255: Reserved Reset type: SYSRSn

13.11.6.2 CONTROL Register (Offset = 4h) [Reset = 00000000h]

CONTROL is shown in Figure 13-39 and described in Table 13-45.

Return to the Summary Table.

Control Register

Figure 13-39. CONTROL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	OVRFLG	RUNSTS	BURSTSTS	TRANSFERSTS	RESERVED	RESERVED	PERINTFLG
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
ERRCLR	RESERVED	RESERVED	PERINTCLR	PERINTFRC	SOFTRESET	HALT	RUN
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 13-45. CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	Reserved
14	OVRFLG	R	0h	Overflow Flag This bit indicates that a peripheral event trigger was received while PERINTFLG was already set. It can be cleared by writing to the ERRCLR bit. Reset type: SYSRSn 0h (R/W) = No overflow detected 1h (R/W) = Overflow detected
13	RUNSTS	R	0h	Run Status Flag This bit indicates that the DMA channel is ready to respond to peripheral event triggers. This bit is set when a 1 is written to the RUN bit. It is cleared when a transfer completes (TRANSFER_COUNT = 0) and corresponding Write access is complete and continuous mode is disabled, or when the HARDRESET, SOFTRESET, or HALT bit is set. Reset type: SYSRSn 0h (R/W) = The channel is disabled 1h (R/W) = The channel is enabled
12	BURSTSTS	R	0h	Burst Status Flag This bit is set when a DMA burst begins. The BURST_COUNT is set to the BURST_SIZE. This bit is cleared when BURST_COUNT reaches zero and corresponding Write access is complete, or when the HARDRESET or SOFTRESET bit is set. Reset type: SYSRSn 0h (R/W) = No burst activity 1h (R/W) = The DMA is currently servicing or suspending a burst transfer from this channel

Table 13-45. CONTROL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	TRANSFERSTS	R	0h	Transfer Status Flag This bit is set when a DMA transfer begins. The address registers are copied to the shadow set and the TRANSFER_COUNT is set to the TRANSFER_SIZE. This bit is cleared when TRANSFER_COUNT reaches zero and corresponding Write access is complete, or when the HARDRESET or SOFTRESET bit is set. Reset type: SYSRSn 0h (R/W) = No transfer activity 1h (R/W) = The channel is currently in the middle of a transfer regardless of whether a burst of data is actively being transferred or not
10	RESERVED	R	0h	Reserved
9	RESERVED	R	0h	Reserved
8	PERINTFLG	R	0h	Peripheral Event Trigger Flag This bit indicates whether a peripheral event trigger has arrived. This bit is automatically cleared when the first burst transfer begins. Reset type: SYSRSn 0h (R/W) = Waiting for event trigger 1h (R/W) = Event trigger pending
7	ERRCLR	R-0/W1S	0h	Clear Error Writing a 1 to this bit will clear the OVRFLG bit. This is normally done when initializing the DMA module or if an overflow condition is detected. If an overflow event occurs at the same time this bit is set, the overrun has priority and the OVRFLG bit is set. [Note] When Overflow and perintflg are set for a channel and at the same time trigger is asserted for the same channel and when its ERRCLR bit is also set to 1, the overflow flag will be de-asserted. Reset type: SYSRSn
6	RESERVED	R-0/W1S	0h	Reserved
5	RESERVED	R-0/W1S	0h	Reserved
4	PERINTCLR	R-0/W1S	0h	Clear Peripheral Event Trigger Writing a 1 to this bit clears PERINTFLG, which cancels a pending event trigger. This is normally done when initializing the DMA module. If an event trigger arrives at the same time this bit is set, the trigger has priority and PERINTFLG is set. Reset type: SYSRSn
3	PERINTFRC	R-0/W1S	0h	Force Peripheral Event Trigger If the PERINTE bit of the MODE register is set, writing a 1 to this bit sets PERINTFLG, which triggers a DMA burst. This bit can be used to start a DMA transfer in software. Reset type: SYSRSn
2	SOFTRESET	R-0/W1S	0h	Channel Soft Reset Writing a 1 to this bit places the channel into its default state after the current read/write access has completed: RUNSTS = 0 TRANSFERSTS = 0 BURSTSTS = 0 BURST_COUNT = 0 TRANSFER_COUNT = 0 SRC_WRAP_COUNT = 0 DST_WRAP_COUNT = 0 When writing to this bit, there is a one cycle delay before it takes effect. Hence, a one-cycle delay (such as a NOP instruction) is required in software before attempting to access any other DMA register. Reset type: SYSRSn

Table 13-45. CONTROL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	HALT	R-0/W1S	0h	Halt Channel Writing a 1 to this bit halts the DMA channel in its current state after any ongoing read/write access has completed. Reset type: SYSRSn
0	RUN	R-0/W1S	0h	Run Channel Writing a 1 to this bit enables the DMA channel and sets the RUNSTS bit to 1. This bit is also used to resume after a channel halt. The RUN bit is typically used to start the DMA channel after configuration. The channel will then wait for the first peripheral event trigger (PERINTFLG == 1) to start a burst. Reset type: SYSRSn

13.11.6.3 BURST_SIZE Register (Offset = 8h) [Reset = 00000000h]

BURST_SIZE is shown in Figure 13-40 and described in Table 13-46.

Return to the Summary Table.

Burst Size Register

Figure 13-40. BURST_SIZE Register

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

Table 13-46. BURST_SIZE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	BURSTSIZE	R/W	0h	These bits specify the burst size in 8-bit words. The actual size is equal to BURSTSIZE + 1. Reset type: SYSRSn

13.11.6.4 BURST_COUNT Register (Offset = Ch) [Reset = 00000000h]

BURST_COUNT is shown in Figure 13-41 and described in Table 13-47.

Return to the Summary Table.

Burst Count Register

Figure 13-41. BURST_COUNT Register

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

Table 13-47. BURST_COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	BURSTCOUNT	R	0h	These bits indicate the number of words left in the current burst. Reset type: SYSRSn 0h (R/W) = 0 word left in a burst 1h (R/W) = 1 word left in a burst 2h (R/W) = 2 word left in a burst 3h (R/W) = 3 word left in a burst 4h (R/W) = 4 word left in a burst 5h (R/W) = 5 word left in a burst 6h (R/W) = 6 word left in a burst 7h (R/W) = 7 word left in a burst 8h (R/W) = 8 word left in a burst 9h (R/W) = 9 word left in a burst Ah (R/W) = 10 word left in a burst Bh (R/W) = 11 word left in a burst Ch (R/W) = 12 word left in a burst Dh (R/W) = 13 word left in a burst Eh (R/W) = 14 word left in a burst Fh (R/W) = 15 word left in a burst 10h (R/W) = 16 word left in a burst 11h (R/W) = 17 word left in a burst 12h (R/W) = 18 word left in a burst 13h (R/W) = 19 word left in a burst 14h (R/W) = 20 word left in a burst 15h (R/W) = 21 word left in a burst 16h (R/W) = 22 word left in a burst 17h (R/W) = 23 word left in a burst 18h (R/W) = 24 word left in a burst 19h (R/W) = 25 word left in a burst 1Ah (R/W) = 26 word left in a burst 1Bh (R/W) = 27 word left in a burst 1Ch (R/W) = 28 word left in a burst 1Dh (R/W) = 29 word left in a burst 1Eh (R/W) = 30 word left in a burst 1Fh (R/W) = 31 word left in a burst

13.11.6.5 SRC_BURST_STEP Register (Offset = 10h) [Reset = 00000000h]

SRC_BURST_STEP is shown in Figure 13-42 and described in Table 13-48.

Return to the Summary Table.

Source Burst Step Register

Figure 13-42. SRC_BURST_STEP Register

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

Table 13-48. SRC_BURST_STEP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	SRCBURSTSTEP	R/W	0h	These bits specify the change in the source address after each word in a burst. The size must be a 16-bit two's complement value between -4096 and 4095 (inclusive). This value is added to the source address after each read/write operation in a burst. Reset type: SYSRSn 0h (R/W) = No address change 1h (R/W) = Add 1 to the address 2h (R/W) = Add 2 to the address FFEh (R/W) = Add 4094 to the address FFFh (R/W) = Add 4095 to the address F000h (R/W) = Subtract 4096 from the address F001h (R/W) = Subtract 4095 from the address FFFEh (R/W) = Subtract 2 from the address FFFFh (R/W) = Subtract 1 from the address

13.11.6.6 DST_BURST_STEP Register (Offset = 14h) [Reset = 00000000h]

DST_BURST_STEP is shown in Figure 13-43 and described in Table 13-49.

Return to the Summary Table.

Destination Burst Step Register

Figure 13-43. DST_BURST_STEP Register

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

Table 13-49. DST_BURST_STEP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	DSTBURSTSTEP	R/W	0h	These bits specify the change in the destination address after each word in a burst. The size must be a 16-bit two's complement value between -4096 and 4095 (inclusive). This value is added to the destination address after each read/write operation in a burst. Reset type: SYSRSn 0h (R/W) = No address change 1h (R/W) = Add 1 to the address 2h (R/W) = Add 2 to the address FFEh (R/W) = Add 4094 to the address FFFh (R/W) = Add 4095 to the address F000h (R/W) = Subtract 4096 from the address F001h (R/W) = Subtract 4095 from the address FFFEh (R/W) = Subtract 2 from the address FFFFh (R/W) = Subtract 1 from the address

13.11.6.7 TRANSFER_SIZE Register (Offset = 18h) [Reset = 00000000h]

TRANSFER_SIZE is shown in Figure 13-44 and described in Table 13-50.

Return to the Summary Table.

Transfer Size Register

Figure 13-44. TRANSFER_SIZE Register

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

Table 13-50. TRANSFER_SIZE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	TRANSFERSIZE	R/W	0h	These bits specify the transfer size in bursts. The actual size is equal to TRANSFERSIZE + 1. Reset type: SYSRSn

13.11.6.8 TRANSFER_COUNT Register (Offset = 1Ch) [Reset = 00000000h]

TRANSFER_COUNT is shown in Figure 13-45 and described in Table 13-51.

Return to the Summary Table.

Transfer Count Register

Figure 13-45. TRANSFER_COUNT Register

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

Table 13-51. TRANSFER_COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	TRANSFERCOUNT	R	0h	These bits indicate the number of bursts left in the current transfer. Reset type: SYSRSn

13.11.6.9 SRC_TRANSFER_STEP Register (Offset = 20h) [Reset = 00000000h]

SRC_TRANSFER_STEP is shown in Figure 13-46 and described in Table 13-52.

Return to the Summary Table.

Source Transfer Step Register

Figure 13-46. SRC_TRANSFER_STEP Register

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

Table 13-52. SRC_TRANSFER_STEP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	SRCTRANSFERSTEP	R/W	0h	These bits specify the change in the source address after a burst completes. The size must be a 16-bit two's complement value between -4096 and 4095 (inclusive). This value is added to the source address after each burst completes. Reset type: SYSRSn 0h (R/W) = No address change 1h (R/W) = Add 1 to the address 2h (R/W) = Add 2 to the address FFEh (R/W) = Add 4094 to the address FFFh (R/W) = Add 4095 to the address F000h (R/W) = Subtract 4096 from the address F001h (R/W) = Subtract 4095 from the address FFFEh (R/W) = Subtract 2 from the address FFFFh (R/W) = Subtract 1 from the address

13.11.6.10 DST_TRANSFER_STEP Register (Offset = 24h) [Reset = 00000000h]

DST_TRANSFER_STEP is shown in Figure 13-47 and described in Table 13-53.

Return to the Summary Table.

Destination Transfer Step Register

Figure 13-47. DST_TRANSFER_STEP Register

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

Table 13-53. DST_TRANSFER_STEP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	DSTTRANSFERSTEP	R/W	0h	These bits specify the change in the destination address after a burst completes. The size must be a 16-bit two's complement value between -4096 and 4095 (inclusive). This value is added to the destination address after each burst completes. Reset type: SYSRSn 0h (R/W) = No address change 1h (R/W) = Add 1 to the address 2h (R/W) = Add 2 to the address FFEh (R/W) = Add 4094 to the address FFFh (R/W) = Add 4095 to the address F000h (R/W) = Subtract 4096 from the address F001h (R/W) = Subtract 4095 from the address FFFEh (R/W) = Subtract 2 from the address FFFFh (R/W) = Subtract 1 from the address

13.11.6.11 SRC_WRAP_SIZE Register (Offset = 28h) [Reset = 0000FFFFh]

SRC_WRAP_SIZE is shown in Figure 13-48 and described in Table 13-54.

Return to the Summary Table.

Source Wrap Size Register

Figure 13-48. SRC_WRAP_SIZE Register

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

Table 13-54. SRC_WRAP_SIZE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	WRAPSIZE	R/W	FFFFh	These bits specify the number of bursts to transfer before the source address wraps around to the beginning address. The actual number is equal to WRAPSIZE + 1. To disable the wrapping function, set WRAPSIZE to a value larger than TRANSFERSIZE. Reset type: SYSRSn

13.11.6.12 SRC_WRAP_COUNT Register (Offset = 2Ch) [Reset = 00000000h]

SRC_WRAP_COUNT is shown in Figure 13-49 and described in Table 13-55.

Return to the Summary Table.

Source Wrap Count Register

Figure 13-49. SRC_WRAP_COUNT Register

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

Table 13-55. SRC_WRAP_COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	WRAPSIZE	R	0h	These bits indicate the number of bursts left before wrapping the source address. Reset type: SYSRSn

13.11.6.13 SRC_WRAP_STEP Register (Offset = 30h) [Reset = 00000000h]

SRC_WRAP_STEP is shown in Figure 13-50 and described in Table 13-56.

Return to the Summary Table.

Source Wrap Step Register

Figure 13-50. SRC_WRAP_STEP Register

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

Table 13-56. SRC_WRAP_STEP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	WRAPSTEP	R/W	0h	These bits specify the change in the source beginning address when the wrap counter reaches zero. The size must be a 16-bit two's complement value between -4096 and 4095 (inclusive). This value is added to the source address when wrapping occurs. Reset type: SYSRSn 0h (R/W) = No address change 1h (R/W) = Add 1 to the address 2h (R/W) = Add 2 to the address FFEh (R/W) = Add 4094 to the address FFFh (R/W) = Add 4095 to the address F000h (R/W) = Subtract 4096 from the address F001h (R/W) = Subtract 4095 from the address FFFEh (R/W) = Subtract 2 from the address FFFFh (R/W) = Subtract 1 from the address

13.11.6.14 DST_WRAP_SIZE Register (Offset = 34h) [Reset = 0000FFFFh]

DST_WRAP_SIZE is shown in Figure 13-51 and described in Table 13-57.

Return to the Summary Table.

Destination Wrap Size Register

Figure 13-51. DST_WRAP_SIZE Register

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

Table 13-57. DST_WRAP_SIZE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	WRAPSIZE	R/W	FFFFh	These bits specify the number of bursts to transfer before the destination address wraps around to the beginning address. The actual number is equal to WRAPSIZE + 1. To disable the wrapping function, set WRAPSIZE to a value larger than TRANSFERSIZE. Reset type: SYSRSn

13.11.6.15 DST_WRAP_COUNT Register (Offset = 38h) [Reset = 00000000h]

DST_WRAP_COUNT is shown in Figure 13-52 and described in Table 13-58.

Return to the Summary Table.

Destination Wrap Count Register

Figure 13-52. DST_WRAP_COUNT Register

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

Table 13-58. DST_WRAP_COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	WRAPSIZE	R	0h	These bits indicate the number of bursts left before wrapping the destination address. Reset type: SYSRSn

13.11.6.16 DST_WRAP_STEP Register (Offset = 3Ch) [Reset = 00000000h]

DST_WRAP_STEP is shown in Figure 13-53 and described in Table 13-59.

Return to the Summary Table.

Destination Wrap Step Register

Figure 13-53. DST_WRAP_STEP Register

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

Table 13-59. DST_WRAP_STEP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	WRAPSTEP	R/W	0h	These bits specify the change in the destination beginning address when the wrap counter reaches zero. The size must be a 16-bit two's complement value between -4096 and 4095 (inclusive). This value is added to the destination address when wrapping occurs. Reset type: SYSRSn 0h (R/W) = No address change 1h (R/W) = Add 1 to the address 2h (R/W) = Add 2 to the address FFEh (R/W) = Add 4094 to the address FFFh (R/W) = Add 4095 to the address F000h (R/W) = Subtract 4096 from the address F001h (R/W) = Subtract 4095 from the address FFFEh (R/W) = Subtract 2 from the address FFFFh (R/W) = Subtract 1 from the address

13.11.6.17 SRC_BEG_ADDR_SHADOW Register (Offset = 40h) [Reset = 00000000h]

SRC_BEG_ADDR_SHADOW is shown in Figure 13-54 and described in Table 13-60.

Return to the Summary Table.

Source Begin Address Shadow Register

Figure 13-54. SRC_BEG_ADDR_SHADOW Register

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

Table 13-60. SRC_BEG_ADDR_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BEGADDR	R/W	0h	Shadow Source Beginning Address At the start of a transfer, the value in this register is loaded into the SRC_BEG_ADDR_ACTIVE register and used as the beginning value for the source address. This register can be safely updated during a transfer. Reset type: SYSRSn

13.11.6.18 SRC_ADDR_SHADOW Register (Offset = 44h) [Reset = 00000000h]

SRC_ADDR_SHADOW is shown in Figure 13-55 and described in Table 13-61.

Return to the Summary Table.

Source Address Shadow Register

Figure 13-55. SRC_ADDR_SHADOW Register

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

Table 13-61. SRC_ADDR_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R/W	0h	Shadow Source Address At the start of a transfer, the value in this register is loaded into the SRC_ADDR_ACTIVE register and used as the value of the source address. This register can be safely updated during a transfer. Reset type: SYSRSn

13.11.6.19 SRC_BEG_ADDR_ACTIVE Register (Offset = 48h) [Reset = 00000000h]

SRC_BEG_ADDR_ACTIVE is shown in Figure 13-56 and described in Table 13-62.

Return to the Summary Table.

Source Begin Address Active Register

Figure 13-56. SRC_BEG_ADDR_ACTIVE Register

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

Table 13-62. SRC_BEG_ADDR_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BEGADDR	R	0h	Active Source Beginning Address If a transfer is ongoing, this register holds the current beginning value for the source address. This address may be updated after wrapping. When a transfer starts, this register is loaded with the shadow address from the SRC_BEG_ADDR_SHADOW register. Reset type: SYSRSn

13.11.6.20 SRC_ADDR_ACTIVE Register (Offset = 4Ch) [Reset = 00000000h]

SRC_ADDR_ACTIVE is shown in Figure 13-57 and described in Table 13-63.

Return to the Summary Table.

Source Address Active Register

Figure 13-57. SRC_ADDR_ACTIVE Register

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

Table 13-63. SRC_ADDR_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R	0h	Active Source Address If a transfer is ongoing, this register holds the current value of the source address. This address may change after a write, a burst, or wrapping. Reset type: SYSRSn

13.11.6.21 DST_BEG_ADDR_SHADOW Register (Offset = 50h) [Reset = 00000000h]

DST_BEG_ADDR_SHADOW is shown in Figure 13-58 and described in Table 13-64.

Return to the Summary Table.

Destination Begin Address Shadow Register

Figure 13-58. DST_BEG_ADDR_SHADOW Register

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

Table 13-64. DST_BEG_ADDR_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BEGADDR	R/W	0h	Shadow Destination Beginning Address At the start of a transfer, the value in this register is loaded into the DST_BEG_ADDR_ACTIVE register and used as the beginning value for the destination address. This register can be safely updated during a transfer. Reset type: SYSRSn

13.11.6.22 DST_ADDR_SHADOW Register (Offset = 54h) [Reset = 00000000h]

DST_ADDR_SHADOW is shown in Figure 13-59 and described in Table 13-65.

Return to the Summary Table.

Destination Address Shadow Register

Figure 13-59. DST_ADDR_SHADOW Register

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

Table 13-65. DST_ADDR_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R/W	0h	Shadow Destination Address At the start of a transfer, the value in this register is loaded into the DST_ADDR_ACTIVE register and used as the value of the destination address. This register can be safely updated during a transfer. Reset type: SYSRSn

13.11.6.23 DST_BEG_ADDR_ACTIVE Register (Offset = 58h) [Reset = 00000000h]

DST_BEG_ADDR_ACTIVE is shown in Figure 13-60 and described in Table 13-66.

Return to the Summary Table.

Destination Begin Address Active Register

Figure 13-60. DST_BEG_ADDR_ACTIVE Register

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

Table 13-66. DST_BEG_ADDR_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BEGADDR	R	0h	Active Destination Beginning Address If a transfer is ongoing, this register holds the current destination value for the source address. This address may be updated after wrapping. When a transfer starts, this register is loaded with the shadow address from the DST_BEG_ADDR_SHADOW register. Reset type: SYSRSn

13.11.6.24 DST_ADDR_ACTIVE Register (Offset = 5Ch) [Reset = 00000000h]

DST_ADDR_ACTIVE is shown in Figure 13-61 and described in Table 13-67.

Return to the Summary Table.

Destination Address Active Register

Figure 13-61. DST_ADDR_ACTIVE Register

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

Table 13-67. DST_ADDR_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R	0h	Active Destination Address If a transfer is ongoing, this register holds the current value of the destination address. This address may change after a write, a burst, or wrapping. Reset type: SYSRSn

13.11.6.25 CHSECLAT1 Register (Offset = 80h) [Reset = 00000000h]

CHSECLAT1 is shown in Figure 13-62 and described in Table 13-68.

Return to the Summary Table.

Channel Security Details Latch Register

Figure 13-62. CHSECLAT1 Register

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

Table 13-68. CHSECLAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Reserved
27-24	STACK	R	0h	This field reflects the CPUID value of last access to the corresponding channel registers (RTDMA_CH_REGS) Reset type: SYSRSn
23-20	RESERVED	R	0h	Reserved
19-16	APILINK	R	0h	This field reflects the APILINK value of last access to the corresponding channel registers (RTDMA_CH_REGS) Reset type: SYSRSn
15-12	RESERVED	R	0h	Reserved
11-8	LINK	R	0h	This field reflects the LINK value of last access to the corresponding channel registers (RTDMA_CH_REGS) Reset type: SYSRSn
7-4	RESERVED	R	0h	Reserved
3-0	ZONE	R	0h	This field reflects the ZONE value of last access to the corresponding channel registers (RTDMA_CH_REGS) Reset type: SYSRSn

13.11.6.26 CHSECLAT2 Register (Offset = 84h) [Reset = 00000000h]

CHSECLAT2 is shown in Figure 13-63 and described in Table 13-69.

Return to the Summary Table.

Channel Security Details Latch Register

Figure 13-63. CHSECLAT2 Register

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

Table 13-69. CHSECLAT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	PRIVID	R	0h	This field reflects the PRIVID value of last access to the corresponding channel registers (RTDMA_CH_REGS) Reset type: SYSRSn
15-10	RESERVED	R	0h	Reserved
9-8	PRIV	R	0h	This field reflects the PRIV value of last access to the corresponding channel registers (RTDMA_CH_REGS) Reset type: SYSRSn
7-1	RESERVED	R	0h	Reserved
0	SECURE	R	0h	This field reflects the SECURE value of last access to the corresponding channel registers (RTDMA_CH_REGS) Reset type: SYSRSn

13.11.6.27 BURST_INTF_CTRL Register (Offset = A0h) [Reset = 00000000h]

BURST_INTF_CTRL is shown in Figure 13-64 and described in Table 13-70.

Return to the Summary Table.

Burst Interface Control Register

Figure 13-64. BURST_INTF_CTRL Register

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

Table 13-70. BURST_INTF_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	BURSTCTRL	R/W	0h	0 : Burst Mode is disabled and the RTDMA accesses data as normal. 1 : Reserved 2 : Burst mode is enabled and cannot be interrupted. All other values are reserved. Reset type: SYSRSn

13.11.6.28 CHCFG_LOCK Register (Offset = 100h) [Reset = 00000000h]

CHCFG_LOCK is shown in Figure 13-65 and described in Table 13-71.

Return to the Summary Table.

Channel Configuration Temporary Lock

Figure 13-65. CHCFG_LOCK Register

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

Table 13-71. CHCFG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	When set, locks this corresponding channel configuration registers (writes will have no effect on them). This bit can only be modified if CHCFG_COMMIT.COMMIT is cleared. 0 : Unlocked 1 : Locked Reset type: SYSRSn

13.11.6.29 CHCFG_COMMIT Register (Offset = 104h) [Reset = 00000000h]

CHCFG_COMMIT is shown in Figure 13-66 and described in Table 13-72.

Return to the Summary Table.

Channel Configuration Permanent Commit

Figure 13-66. CHCFG_COMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 13-72. CHCFG_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the CHCFG_LOCK register. This bit cannot be cleared, except by reset. 0 : CHCFG_LOCK is modifiable 1 : CHCFG_LOCK is committed permanently Reset type: SYSRSn

Chapter 14

External Memory Interface (EMIF)

This chapter describes the external memory interface (EMIF).

Further information can be found in the following documents:

Accessing External SDRAM on the TMS320F2837x/2807x Microcontrollers Using C/C++ Application Report

Design and Usage Guidelines for the C2000™ External Memory Interface (EMIF) Application Report

14.1 Introduction

This device supports one EMIF module — EMIF1, as shown in Figure 14-1.

Table 14-1 gives the configuration for the EMIF module.

The EMIF module is accessible from all CPUs, but fast access ports are limited to the primary CPU. Other initiators use the slow access ports. Code execution from the EMIF is not supported.

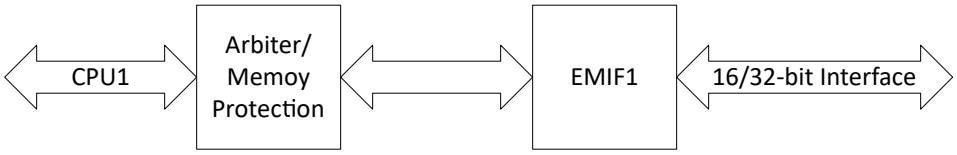


Figure 14-1. EMIF Module Overview

Table 14-1. Configuration for the EMIF1 Module

	EMIF1
Maximum Data Width	32
Maximum Address Width	22 (Some of EMIF1 pins are muxed with each other. Refer to Section 14.2.11 for usage)
SDRAM CSx Support	1 (CS0)
ASRAM CSx Support	3 (CS2/CS3/CS4)

Note

Subsequent sections in this chapter provide the details on the generic EMIF module. Unless otherwise specified, pin names are used from EMIF1 to define the functionality.

14.1.1 Purpose of the Peripheral

This EMIF memory controller is compliant with the JESD21-C SDR SDRAM memories utilizing a 32-bit/16-bit data bus. The purpose of this EMIF is to provide a means for the CPU to connect to a variety of external devices including:

- Single data rate (SDR) SDRAM
- Asynchronous devices including NOR Flash and SRAM

A common use for the EMIF is to interface with both a Flash device and an SDRAM device simultaneously. Section 14.4 contains an example of operating the EMIF in this configuration.

14.1.2 Features

The EMIF controller includes many features to enhance the ease and flexibility of connecting to the external SDR SDRAM and asynchronous devices.

- Accessible by all CPUs, RTDMA1, and RTDMA2
  - Fast port access from primary CPU, slow port access from other CPUs and RTDMA
- RTDMA burst support
- Separate buffer module for each CPU with a write FIFO that contains up to 4 entries
- EMIF splits accesses based on data size
  - 64-bit access causes two 32-bit accesses
- Arbitration for same and different initiators
- Supports single chip select for SDRAM, three chip selects for ASRAM, and an additional range for EMIF register accesses

14.1.2.1 Asynchronous Memory Support

The EMIF controller supports asynchronous:

- SRAM memories
- NOR Flash memories

There is an external wait input that allows slower asynchronous memories to extend the memory access. The EMIF module supports more than one chip select (enable). Each chip select has the following individually programmable attributes:

- Data bus width
- Read cycle timings: setup, hold, strobe
- Write cycle timings: setup, hold, strobe
- Bus turnaround time
- Extended wait option with programmable timeout
- Select strobe option

14.1.2.2 Synchronous DRAM Memory Support

The EMIF module supports 16-bit/32-bit SDRAM in addition to the asynchronous memories listed in Section 14.1.2.1. The EMIF module has a single SDRAM chip select. SDRAM configurations that are supported are:

- One, two and four bank SDRAM devices
- Devices with eight, nine, ten, and eleven column address
- CAS latency of two or three clock cycles
- 16-bit/32-bit data bus width
- 3.3V LVCMOS interface

Additionally, the EMIF supports placing the SDRAM in self-refresh and power-down modes. The self-refresh mode allows the SDRAM to be put in a low-power state while still retaining memory contents, since the SDRAM continues to refresh itself even without clocks from the microcontroller. The power-down mode achieves even lower power, except the microcontroller must periodically wake up and issue refreshes if data retention is required.

Note that the EMIF module does not support mobile SDRAM devices.

14.1.3 Functional Block Diagram

Figure 14-2 shows the connections between the EMIF and the internal requesters, along with the external EMIF pins. Section 14.2.2 contains a description of the entities internal to the MCU that can send requests to the EMIF, along with their prioritization. Section 14.2.3 describes the EMIF external pins and summarizes their purpose when interfacing with the SDRAM and asynchronous devices.

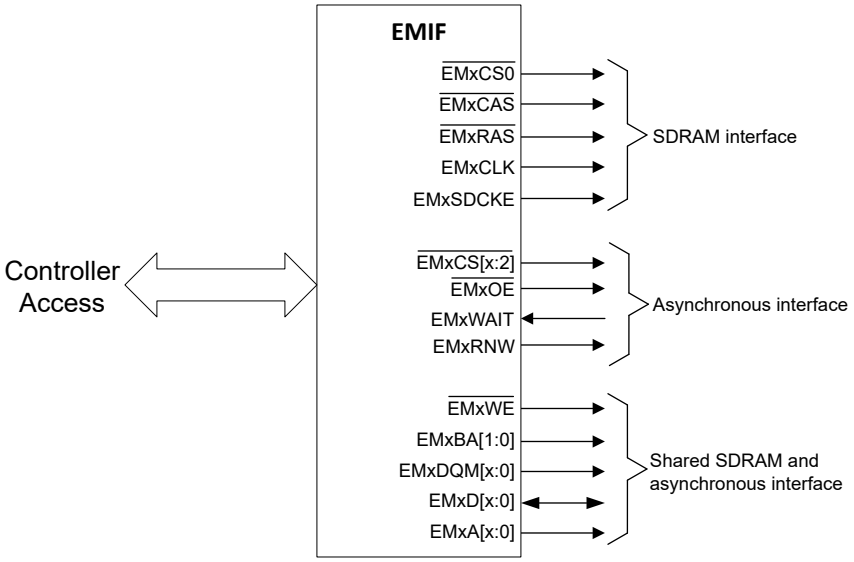


Figure 14-2. EMIF Functional Block Diagram

14.1.4 Configuring Device Pins

The GPIO mux registers must be configured to connect this peripheral to the device pins. To avoid glitches on the pins, the GPyGMUX bits must be configured first (while keeping the corresponding GPyMUX bits at the default of zero), followed by writing the GPyMUX register to the desired value.

Some IO functionality is defined by GPIO register settings independent of this peripheral. For input signals, the GPIO input qualification must be set to asynchronous mode by setting the appropriate GPxQSELn register bits to 11b. The internal pullups can be configured in the GPyPUD register.

See the *General-Purpose Input/Output (GPIO)* chapter for more details on GPIO mux and settings.

14.2 EMIF Module Architecture

This section provides details about the architecture and operation of the EMIF. Both the SDRAM and asynchronous interface are covered, along with other system-related configurations such as clock control.

14.2.1 EMIF Clock Control

The EMIF clock is output on the pin and must be used when interfacing to external SDRAM devices. The EMIF module gets the PLLSYSCLK clock domain as the input. The user can choose to run the EMIF at PLLSYSCLK/1 or PLLSYSCLK/2 clock frequency by configuring the field in the PERCLKDIVSEL register in the *Clock Control* module.

14.2.2 EMIF Requests

Different sources within the MCU can make requests to the EMIF. These requests consist of accesses to the SDRAM memory, the asynchronous memory, and the EMIF registers. The EMIF can process only one request at a time. Therefore, a high performance controller arbitration block exists within the MCU to provide prioritized requests from the different sources to the EMIF. The sources are:

- CPU1
- CPU2
- CPU3
- RTDMA

If a request is submitted from two or more sources simultaneously, the crossbar switch forwards the highest priority request to the EMIF first. Upon completion of a request, the controller arbitration block again evaluates the pending requests and forwards the highest priority pending request to the EMIF.

The controller arbitration block always allows RD access from any of the controller. But for WR access (or execute access), the controller arbitration block arbitrates between the accesses controller from multiple resources (CPU1/2/3 and RTDMA) controller using a round robin priority scheme.

When the EMIF receives a request, it is possible that the request is not immediately processed. In some cases, the EMIF performs one or more auto-refresh cycles before processing the request. For details on the EMIF internal arbitration between performing requests and performing auto-refresh cycles, see Section 14.2.13.

14.2.3 EMIF Signal Descriptions

This section describes the function of each of the EMIF signals.

Table 14-2. EMIF Pins Used to Access Both SDRAM and Asynchronous Memories

Pins	I/O	Description
EM1D[x:0]	I/O	**EMIF data bus.**
EM1A[x:0]	O	**EMIF address bus.** When interfacing to an SDRAM device, these pins are primarily used to provide the row and column address to the SDRAM. The mapping from the internal program address to the external values placed on these pins is found in Table 14-14. EM1A[10] is also used during the PRE command to select which banks to deactivate. When interfacing to an asynchronous device, these pins are used in conjunction with the EM1BA pins to form the address that is sent to the device. The mapping from the internal program address to the external values placed on these pins is found in Section 14.2.6.1.
EM1BA[1:0]	O	**EMIF bank address.** When interfacing to an SDRAM device, these pins are used to provide the bank address inputs to the SDRAM. The mapping from the internal program address to the external values placed on these pins is found in Table 14-14. When interfacing to an asynchronous device, these pins are used in conjunction with the EM1A pins to form the address that is sent to the device. The mapping from the internal program address to the external values placed on these pins is found in Section 14.2.6.1.
EM1DQM[x:0]	O	**Active-low byte enables.** When interfacing to SDRAM, these pins are connected to the DQM pins of the SDRAM to individually enable/disable each of the bytes in a data access. When interfacing to an asynchronous device, these pins are connected to byte enables. See Section 14.2.6 for details.
$\overline{\text{EM1WE}}$	O	**Active-low write enable.** When interfacing to SDRAM, this pin is connected to the nWE pin of the SDRAM and is used to send commands to the device. When interfacing to an asynchronous device, this pin provides a signal which is active-low during the strobe period of an asynchronous write access cycle.

Table 14-3. EMIF Pins Specific to SDRAM

Pins	I/O	Description
$\overline{\text{EM1CS0}}$	O	**Active-low chip enable pin for SDRAM devices.** This pin is connected to the chip-select pin of the attached SDRAM device and is used for enabling/disabling commands. By default, EMIF keeps this SDRAM chip select active, even if EMIF is not interfaced with an SDRAM device. This pin is deactivated when accessing the asynchronous memory bank and is reactivated on completion of the asynchronous access.
EM1RAS	O	**Active-low row address strobe pin.** This pin is connected to the nRAS pin of the attached SDRAM device and is used for sending commands to the device.
EM1CAS	O	**Active-low column address strobe pin.** This pin is connected to the nCAS pin of the attached SDRAM device and is used for sending commands to the device.
EM1SDCKE	O	**Clock enable pin.** This pin is connected to the CKE pin of the attached SDRAM device and is used for issuing the SELF REFRESH command which places the device in self-refresh mode. See Section 14.2.5.7 for details.
EM1CLK	O	**SDRAM clock pin.** This pin is connected to the CLK pin of the attached SDRAM device. See Section 14.2.1 for details on the clock signal.

Table 14-4. EMIF Pins Specific to Asynchronous Memory

Pins	I/O	Description
$\overline{\text{EM1CS}}$[4:2]	O	**Active-low chip enable pins for asynchronous devices.** These pins are meant to be connected to the chip-select pins of the attached asynchronous device. These pins are active only during accesses to the asynchronous memory.
EM1WAIT	I	**Wait input with programmable polarity.** A connected asynchronous device can extend the strobe period of an access cycle by asserting the EM1WAIT input to EMIF as described in Section 14.2.6.6. To enable this functionality, the EW bit in the asynchronous 1 configuration register (ASYNC_CS2_CFG) must be set to 1. In addition, the WP0 bit in ASYNC_CS2_CFG must be configured to define the polarity of the EM1WAIT pin.
$\overline{\text{EM1OE}}$	O	**Active-low pin enable for asynchronous devices.** This pin provides a signal which is active-low during the strobe period of an asynchronous read access cycle.
EM1RNW	O	**EMIF asynchronous read/write control.** This pin stays high during reads and stays low during writes (same duration as CS).

14.2.4 EMIF Signal Multiplexing Control

Several EMIF signals are multiplexed with other functions on this microcontroller. Refer to the multiplexing section of the *General-Purpose Input/Output (GPIO)* chapter for more information on how to enable the output of these EMIF signals.

14.2.5 SDRAM Controller and Interface

The EMIF controller provides a glueless interface to most standard SDR SDRAM devices and supports features like self-refresh mode and prioritized refresh. In addition, the EMIF controller provides flexibility through programmable parameters such as the refresh rate, CAS latency, and many SDRAM timing parameters. The following sections include details on how to Interface and properly configure the EMIF to perform read and write operations to externally connected SDR SDRAM devices. Also, Section 14.4 provides a detailed example of interfacing the EMIF to a common SDRAM device.

14.2.5.1 SDRAM Commands

The EMIF controller supports the SDRAM commands described in Table 14-5. Table 14-6 shows the truth table for the SDRAM commands, and an example timing waveform of the PRE command is shown in Figure 14-3. EM1A[10] is pulled low in this example to deactivate only the bank specified by the EM1BA pins.

Table 14-5. EMIF SDRAM Commands

Command	Function
PRE	**Precharge.** Depending on the value of EM1A[10], the PRE command either deactivates the open row in all banks (EM1A[10] = 1) or only the bank specified by the EM1BA[1:0] pins (EM1A[10] = 0).
ACTV	**Activate**. The ACTV command activates the selected row in a particular bank for the current access.
READ	**Read.** The READ command outputs the starting column address and signals the SDRAM to begin the burst read operation. Address EM1A[10] is always pulled low to avoid auto precharge. This allows for better bank interleaving performance.
WRT	**Write.** The WRT command outputs the starting column address and signals the SDRAM to begin the burst write operation. Address EM1A[10] is always pulled low to avoid auto precharge. This allows for better bank interleaving performance.
BT	**Burst terminate**. The BT command is used to truncate the current read or write burst request. On this device, all the SDRAM accesses are single access except when EMIF controller splits a single access into multiple access (for example, a 32-bit access from CPU is split into two 16-bit accesses if external SDRAM device is 16 bit (NM =1).
LMR	**Load mode register**. The LMR command sets the mode register of the attached SDRAM devices and is only issued during the SDRAM initialization sequence described in Section 14.2.5.4.
REFR	**Auto refresh**. The REFR command signals the SDRAM to perform an auto refresh according to the internal address.
SLFR	**Self refresh**. The self-refresh command places the SDRAM into self-refresh mode, during which the SDRAM provides a clock signal and auto refresh cycles.
NOP	**No operation**. The NOP command is issued during all cycles in which one of the above commands is not issued.

Table 14-6. Truth Table for SDRAM Commands

SDRAM Pins:	CKE	nCS	nRAS	nCAS	nWE	BA[1:0]	A[12:11]	A[10]	A[9:0]
EMIF Pins:	EM1SDCKE	$\overline{\text{EM1CS}}$[0]	EM1RAS	EM1CAS	$\overline{\text{EM1WE}}$	EM1BA[1:0]	EM1A[12:11]	EM1A[10]	EM1A[9:0]
PRE	H	L	L	H	L	Bank/X	X	L/H	X
ACTV	H	L	L	H	H	Bank	Row	Row	Row
READ	H	L	H	L	H	Bank	Column	L	Column
WRT	H	L	H	L	L	Bank	Column	L	Column
BT	H	L	H	H	L	X	X	X	X
LMR	H	L	L	L	L	X	Mode	Mode	Mode
REFR	H	L	L	L	H	X	X	X	X
SLFR	L	L	L	L	H	X	X	X	X
NOP	H	L	H	H	H	X	X	X	X

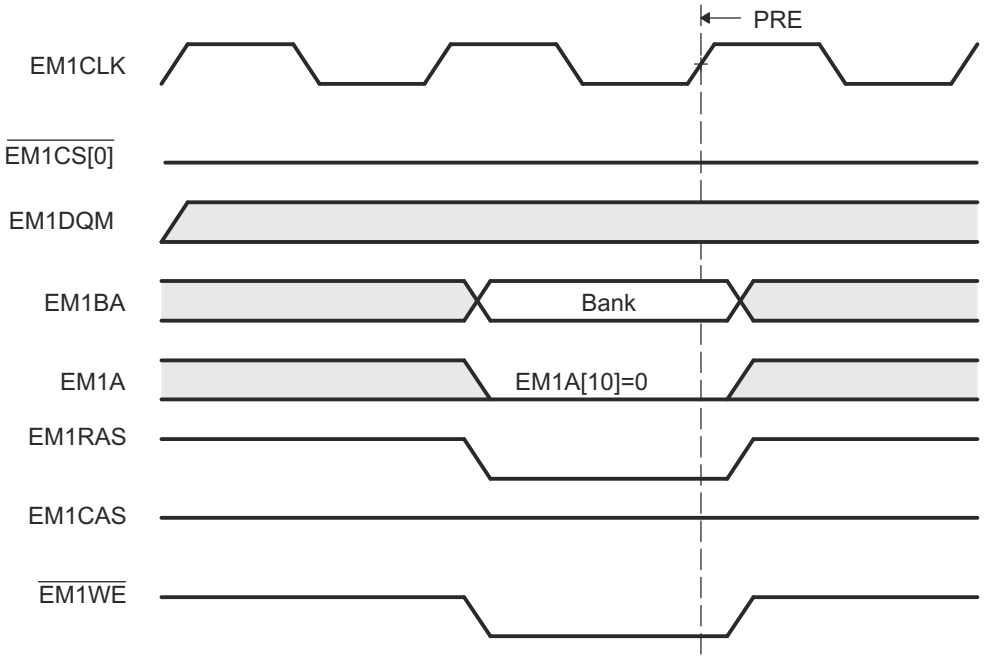


Figure 14-3. Timing Waveform of SDRAM PRE Command

14.2.5.2 Interfacing to SDRAM

The EMIF supports a glueless interface to SDRAM devices with the following characteristics:

- Pre-charge bit is A[10]
- The number of column address bits is 8, 9, 10, or 11.
- The number of row address bits is 13, 14, 15, or 16.
- The number of internal banks is 1, 2, or 4.

Figure 14-4 shows an interface between the EMIF and a 2M × 16 × 4 bank SDRAM device, and Figure 14-5 shows an interface between the EMIF and a 512K × 16 × 2 bank SDRAM device. For devices supporting 16-bit interface, refer to Table 14-7 for list of commonly-supported SDRAM devices and the required connections for the address pins.

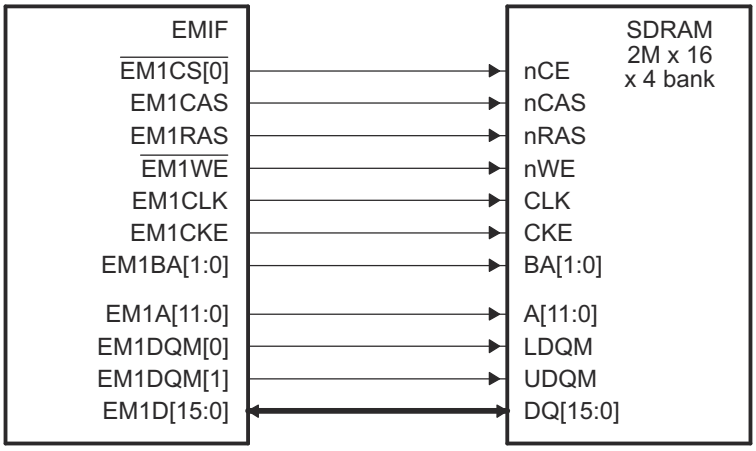


Figure 14-4. EMIF to 2M × 16 × 4 Bank SDRAM Interface

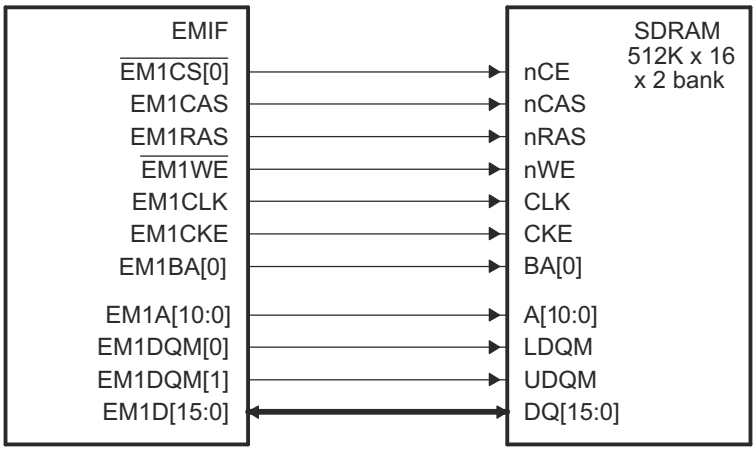


Figure 14-5. EMIF to 512K × 16 × 2 Bank SDRAM Interface

Table 14-7. 16-bit EMIF Address Pin Connections

SDRAM Size	Width	Banks	Device	Address Pins
16M bits	×16	2	SDRAM	A[10:0]
			EMIF	EM1A[10:0]
64M bits	×16	4	SDRAM	A[11:0]
			EMIF	EM1A[11:0]
128M bits	×16	4	SDRAM	A[11:0]
			EMIF	EM1A[11:0]
256M bits	x16	4	SDRAM	A[12:0]
			EMIF	EM1A[12:0]
512M bits	x16	4	SDRAM	A[12:0]
			EMIF	EM1A[12:0]

14.2.5.3 SDRAM Configuration Registers

The operation of the EMIF SDRAM interface is controlled by programming the appropriate configuration registers. This section describes the purpose and function of each configuration register, but Section 14.6 can be referred to for a more detailed description of each register, including the default registers values and bit-field positions. The following tables list the four such configuration registers, along with a description of each of their programmable fields.

Note

Writing to any of the fields: NM, CL, IBANK, and PAGESIZE in the SDRAM configuration register (SDRAM_CR) causes the EMIF to abandon whatever the EMIF is currently doing and trigger the SDRAM initialization procedure described in Section 14.2.5.4.

Table 14-8. Description of the SDRAM Configuration Register (SDRAM_CR)

Parameter	Description
SR	This bit controls entering and exiting of the self-refresh mode
PD	This bit controls entering and exiting of the power-down mode. If both SR and PD bits are set, the EMIF goes into self-refresh mode.
PDWR	Perform refreshes during power down. Writing a 1 to this bit causes the EMIF to exit the power-down state and issue an AUTO REFRESH command every time Refresh May level is set. This bit must be set along with PD when entering power-down mode.
NM	**Narrow Mode.** This bit defines the width of the data bus between the EMIF and the attached SDRAM device. When set to 1, the data bus is set to 16-bits. When set to 0, the data bus is set to 32-bits.
CL	**CAS latency.** This field defines the number of clock cycles between when an SDRAM issues a READ command and when the first piece of data appears on the bus. The value in this field is sent to the attached SDRAM device using the LOAD MODE REGISTER command during the SDRAM initialization procedure as described in Section 14.2.5.4. Only, values of 2h (CAS latency = 2) and 3h (CAS latency = 3) are supported and must be written to this field. While updating the CL field, BIT11_9LOCK bit field must be set to 1 simultaneously.
IBANK	**Number of Internal SDRAM Banks.** This field defines the number of banks inside the attached SDRAM devices in the following way: • When IBANK = 0, 1 internal bank is used • When IBANK = 1h, 2 internal banks are used • When IBANK = 2h, 4 internal banks are used This field value affects the mapping of logical addresses to the SDRAM row, column, and bank addresses. See Section 14.2.5.11 for details.
PAGESIZE	**Page Size.** This field defines the internal page size of the attached SDRAM devices in the following way: • When PAGESIZE = 0, 256-word pages are used • When PAGESIZE = 1h, 512-word pages are used • When PAGESIZE = 2h, 1024-word pages are used • When PAGESIZE = 3h, 2048-word pages are used This field value affects the mapping of logical addresses to the SDRAM row, column, and bank addresses. See Section 14.2.5.11 for details.

Table 14-9. Description of the SDRAM Refresh Control Register (SDRAM_RCR)

Parameter	Description
RR	**Refresh Rate**. This field controls the rate at which attached SDRAM devices are refreshed. The following equation can be used to determine the required value of RR for an SDRAM device: • RR = f_{EM1CLK} / (Required SDRAM Refresh Rate) More information about the operation of the SDRAM refresh controller is found in Section 14.2.5.6.

Table 14-10. Description of the SDRAM Timing Register (SDRAM_TR)

Parameter	Description
T_RFC T_RP T_RCD T_WR T_RAS T_RC T_RRD	**SDRAM Timing Parameters.** These fields configure the EMIF to comply with the AC timing requirements of the attached SDRAM devices. This allows the EMIF to avoid violating SDRAM timing constraints and to more efficiently schedule operations. More details about each of these parameters can be found in the SDRAM_TR register description. These parameters must be set to satisfy the corresponding timing requirements found in the SDRAM data sheet.

Table 14-11. Description of the SDRAM Self Refresh Exit Timing Register (SDR_EXT_TMNG)

Parameter	Description
T_XS	**Self Refresh Exit Parameter.** The T_XS field of this register informs the EMIF about the minimum number of EM1CLK cycles required between exiting self-refresh and issuing any command. This parameter must be set to satisfy the t_{XSR} value for the attached SDRAM device.

14.2.5.4 SDRAM Auto-Initialization Sequence

The EMIF automatically performs an SDRAM initialization sequence, regardless of whether the EMIF is interfaced to an SDRAM device, when either of the following two events occur:

- The EMIF comes out of reset. No memory accesses to the SDRAM and asynchronous interfaces are performed until this auto-initialization is complete.
- A write is performed to any of the three least-significant bytes of the SDRAM configuration register (SDRAM_CR)

An SDRAM initialization sequence consists of the following steps:

1. If the initialization sequence is activated by a write to SDRAM_CR, and if any of the SDRAM banks are open, the EMIF issues a PRE command with EM1A[10] held high to indicate all banks. This is done so that the maximum ACTV to PRE timing for an SDRAM is not violated.
2. The EMIF drives EM1SDCKE high and begins continuously issuing NOP commands until eight SDRAM refresh intervals have elapsed. An SDRAM refresh interval is equal to the value of the RR field of the SDRAM refresh control register (SDRAM_RCR), divided by the frequency of EM1CLK (RR/f_{EM1CLK}). This step is used to avoid violating the power-up constraint of most SDRAM devices that requires 200µs (sometimes 100µs) between receiving stable Vdd and CLK and the issuing of a PRE command. Depending on the frequency of EM1CLK, this step can be insufficient to avoid violating the SDRAM constraint. See Section 14.2.5.5 for more information.
3. After the refresh intervals have elapsed, the EMIF issues a PRE command with EM1A[10] held high to indicate all banks.
4. The EMIF issues eight AUTO REFRESH commands.
5. The EMIF issues the LMR command with the EM1A[9:0] pins set as described in Table 14-12.
6. Finally, the EMIF performs a refresh cycle, which consists of the following steps:
   a. Issuing a PRE command with EM1A[10] held high if any banks are open
   b. Issuing an REF command

Table 14-12. SDRAM LOAD MODE REGISTER Command

EM1A[9:7]	EM1A[6:4]	EM1A[3]	EM1A[2:0]
0 (Write bursts are of the programmed burst length in EM1A[2:0])	These bits control the CAS latency of the SDRAM and are set according to CL field in the SDRAM configuration register (SDRAM_CR) as follows: • If CL = 2, EM1A[6:4] = 2h (CAS latency = 2) • If CL = 3, EM1A[6:4] = 3h (CAS latency = 3)	0 (Sequential Burst Type. Interleaved Burst Type not supported)	These bits control the burst length of the SDRAM and are set according to the NM field in the SDRAM configuration register (SDRAM_CR) as follows: • If NM = 0, EM1A[2:0] = 2h (Burst Length = 4) • If NM = 1, EM1A[2:0] = 3h (Burst Length = 8)

14.2.5.5 SDRAM Configuration Procedure

There are two different SDRAM configuration procedures. Although the EMIF automatically performs the SDRAM initialization sequence described in Section 14.2.5.4 when coming out of reset, follow one of the procedures before performing any EMIF memory requests.

Procedure A must be followed if the SDRAM power-up constraint was not violated during the SDRAM auto-initialization sequence detailed in Section 14.2.5.4 on coming out of Reset. The SDRAM power-up constraint specifies that 200µs (sometimes 100µs) must exist between receiving stable Vdd and CLK and the issuing of a PRE command.

Procedure B must be followed if the SDRAM power-up constraint was violated. The 200µs (100µs) SDRAM power-up constraint is violated, if the frequency of EM1CLK is greater than 50MHz (100MHz for 100µs SDRAM power-up constraint) during SDRAM Auto-Initialization Sequence. Procedure B must be followed if there is any doubt that the power-up constraint was not met.

Procedure A — Following is the procedure to be followed if the SDRAM power-up constraint was not violated:

1. Place the SDRAM into self-refresh mode by setting the SR bit of SDRAM_CR to 1. The SDRAM can be placed into self-refresh mode when changing the frequency of the EM1CLK to avoid incurring the 200µs power-up constraint again.
2. Configure the desired EMIF1 clock (EM1CLK) frequency. The frequency of the memory clock must meet the timing requirements in the SDRAM manufacturer's documentation and the timing limitations shown in the electrical specifications of the device data sheet.
3. Remove the SDRAM from self-refresh mode by clearing the SR bit of the SDRAM_CR to 0.
4. Program SDRAM_TR and SDR_EXT_TMNG to satisfy the timing requirements for the attached SDRAM device. The timing parameters must be taken from the SDRAM data sheet.
5. Program the RR field of SDRAM_RCR to match that of the attached device's refresh interval. See Section 14.2.5.6.1 details on determining the appropriate value.
6. Program the SDRAM_CR to match the characteristics of the attached SDRAM device. This causes the auto-initialization sequence in Section 14.2.5.4 to be re-run. This second initialization generally takes much less time due to the increased frequency of EM1CLK.

Procedure B — Following is the procedure to be followed if the SDRAM power-up constraint was violated:

1. Configure the desired EM1CLK clock frequency. The frequency of the memory clock must meet the timing requirements in the SDRAM manufacturer's documentation and the timing limitations shown in the electrical specifications of the device data sheet.
2. Program SDRAM_TR and SDR_EXT_TMNG to satisfy the timing requirements for the attached SDRAM device. The timing parameters must be taken from the SDRAM data sheet.
3. Program the RR field of the SDRAM_RCR such that the following equation is satisfied: $(RR \times 8)/(f_{EM1CLK}) > 200\mu s$ (sometimes 100µs). For example, an EM1CLK frequency of 100MHz requires setting RR to 2501 (9C5h) or higher to meet a 200µs constraint.
4. Program the SDRAM_CR to match the characteristics of the attached SDRAM device. This causes the auto-initialization sequence in Section 14.2.5.4 to be re-run with the new value of RR.
5. Perform a read from the SDRAM to make sure that step 5 of this procedure occurs after the initialization process has completed. Alternatively, wait for 200µs instead of performing a read.
6. Finally, program the RR field to match that of the attached device's refresh interval. See Section 14.2.5.6.1 details on determining the appropriate value.

After following the above procedure, the EMIF is ready to perform accesses to the attached SDRAM device.

14.2.5.6 EMIF Refresh Controller

An SDRAM device requires that each of the rows be refreshed at a minimum required rate. The EMIF can meet this constraint by performing auto refresh cycles at or above this required rate. An auto-refresh cycle consists of issuing a PRE command to all banks of the SDRAM device followed by issuing a REFR command. To inform the EMIF of the required rate for performing auto refresh cycles, the RR field of the SDRAM refresh control register (SDRAM_RCR) must be programmed. The EMIF uses this value along with two internal counters to automatically perform auto refresh cycles at the required rate. The auto-refresh cycles cannot be disabled, even if the EMIF is not interfaced with an SDRAM. The remainder of this section details the EMIF's refresh scheme and provides an example for determining the appropriate value to place in the RR field of the SDRAM_RCR.

The two counters used to perform auto-refresh cycles are a 13-bit refresh interval counter and a 4-bit refresh backlog counter. At reset and upon writing to the RR field, the refresh interval counter is loaded with the value from RR field and begins decrementing, by one, each EMIF clock cycle. When the refresh interval counter reaches zero, the following actions occur:

- The refresh interval counter is reloaded with the value from the RR field and restarts decrementing.
- The 4-bit refresh backlog counter increments unless the counter has already reached the maximum value.

The refresh backlog counter records the number of auto refresh cycles that the EMIF currently has outstanding. This counter is decremented by one each time an auto refresh cycle is performed and incremented by one each time the refresh interval counter expires. The refresh backlog counter saturates at the values of 0000b and 1111b. The EMIF uses the refresh backlog counter to determine the urgency with which an auto refresh cycle must be performed. The four levels of urgency are described in Table 14-13. This refresh scheme allows the required refreshes to be performed with minimal impact on access requests.

Table 14-13. Refresh Urgency Levels

Urgency Level	Refresh Backlog Counter Range	Action Taken
Refresh May	1-3	An auto-refresh cycle is performed only if the EMIF has no requests pending and none of the SDRAM banks are open.
Refresh Release	4-7	An auto-refresh cycle is performed if the EMIF has no requests pending, regardless of whether any SDRAM banks are open.
Refresh Need	8-11	An auto-refresh cycle is performed at the completion of the current access unless there are read requests pending.
Refresh Must	12-15	Multiple auto-refresh cycles are performed at the completion of the current access until the Refresh Release urgency level is reached. At that point, the EMIF can begin servicing any new read or write requests.

14.2.5.6.1 Determining the Appropriate Value for the RR Field

The value that must be programmed into the RR field of the SDRAM_RCR must can be calculated by using the frequency of the EM1CLK signal (f_{EM1CLK}) and the required refresh rate of the SDRAM ($f_{Refresh}$). The following formula can be used:

$RR = f_{EM1CLK} / f_{Refresh}$

The SDRAM data sheet often communicates the required SDRAM Refresh Rate in terms of the number of REFR commands required in a given time interval. The required SDRAM Refresh Rate in the formula above can therefore be calculated by dividing the number of required cycles per time interval (n_{cycles}) by the time interval given in the data sheet ($t_{Refresh\ Period}$) :

$f_{Refresh} = n_{cycles} / t_{Refresh\ Period}$

Combining these formulas, the value that must be programmed into the RR field can be computed as:

$RR = f_{EM1CLK} \times t_{Refresh\ Period} / n_{cycles}$

The following example illustrates calculating the value of RR. Given that:

- f_{EM1CLK} = 100MHz (frequency of EMIF clock)
- $t_{Refresh\ Period}$ = 64ms (required refresh interval of the SDRAM)
- n_{cycles} = 8192 (number of cycles in a refresh interval for the SDRAM)

RR can be calculated as:

RR = 100MHz × 64ms/8192

RR = 781.25

RR = 782 cycles = 30Eh cycles

14.2.5.7 Self-Refresh Mode

The EMIF can be programmed to enter the self-refresh state by setting the SR bit of SDRAM_CR to 1. This causes the EMIF to issue the SLFR command after completing any outstanding SDRAM access requests and clearing the refresh backlog counter by performing one or more auto refresh cycles. This places the attached SDRAM device into self-refresh mode in which the EMIF consumes a minimal amount of power while performing the refresh cycles.

While in the self-refresh state, the EMIF continues to service asynchronous bank requests and register accesses as normal, with one caveat. The EMIF does not park the data bus following a read to asynchronous memory while in the self-refresh state. Instead, the EMIF tri-states the data bus. Therefore, it is not recommended to perform asynchronous read operations while the EMIF is in the self-refresh state to prevent floating inputs on the data bus. More information about data bus parking can be found in Section 14.2.7.

The EMIF exits from the self-refresh state, if either of the following events occur:

- The SR bit of SDRAM_CR is cleared to 0.
- An SDRAM accesses is requested.

The EMIF exits from the self-refresh state by driving EM1SDCKE high and performing an auto refresh cycle.

The attached SDRAM device must also be placed into self-refresh mode when changing the frequency of EM1CLK. If the frequency of EM1CLK changes while the SDRAM is not in self-refresh mode, Procedure B in Section 14.2.5.5 must be followed to reinitialize the device.

14.2.5.8 Power-Down Mode

To support low-power modes, the EMIF can be requested to issue a POWER DOWN command to the SDRAM by setting the PD bit in the SDRAM configuration register (SDRAM_CR). When this bit is set, the EMIF continues normal operation until all outstanding memory access requests have been serviced and the SDRAM refresh backlog (if there is one) has been cleared. At this point, the EMIF enters the power-down state. Upon entering this state, the EMIF issues a POWER DOWN command (same as a NOP command but driving the EM1SDCKE low on the same cycle). The EMIF then maintains the EM1SDCKE low until the EMIF exits the power-down state.

Since the EMIF services the refresh backlog before the EMIF enters the power-down state, all internal banks of the SDRAM are closed (precharged) prior to issuing the POWER DOWN command. Therefore, the EMIF only supports precharge power-down. The EMIF does not support active power-down, where internal banks of the SDRAM are open (active) before the POWER DOWN command is issued.

During the power-down state, the EMIF services the SDRAM, asynchronous memory, and register accesses as normal, returning to the power-down state upon completion.

The PDWR bit in the SDRAM_CR indicates whether the EMIF must perform refreshes in power-down state. If the PDWR bit is set, the EMIF exits the power-down state every time the Refresh Must level is set, performs AUTO REFRESH commands to the SDRAM, and returns back to the power-down state. This evenly distributes the refreshes to the SDRAM in power-down state. If the PDWR bit is not set, the EMIF does not perform any refreshes to the SDRAM. Therefore, the data integrity of the SDRAM is not maintained upon power-down exit, if the PDWR bit is not set.

If the PD bit is cleared while in the power-down state, the EMIF comes out of the power-down state. The EMIF:

- Drives EM1SDCKE high
- Enters the idle state

14.2.5.9 SDRAM Read Operation

When the EMIF receives a read request to the SDRAM from one of the requesters listed in Section 14.2.2, the EMIF performs one or more read access cycles. A read access cycle begins with the issuing of the ACTV command to select the desired bank and row of the SDRAM device. After the row has been opened, the EMIF proceeds to issue a READ command while specifying the desired bank and column address. EM1A[10] is held low during the READ command to avoid auto-precharging. The READ command signals the SDRAM device to output data from the specified address while EMIF issues NOP commands. Following a READ command, the CL field of the SDRAM configuration register (SDRAM_CR) defines how many delay cycles are present before the read data appears on the data bus. This is referred to as the CAS latency.

Figure 14-6 shows the signal waveforms for a basic SDRAM read operation in which multiple data is read from a single page. On this device, burst accesses are not supported; hence, the EMIF issues a READ command for each data access. Only when the EMIF SDRAM interface is configured to 16-bit by setting the NM bit of the SDRAM configuration register (SDRAM_CR) to 1 and CPU (or any other controller) does a 32-bit READ access, a burst access is issued with a size of two.

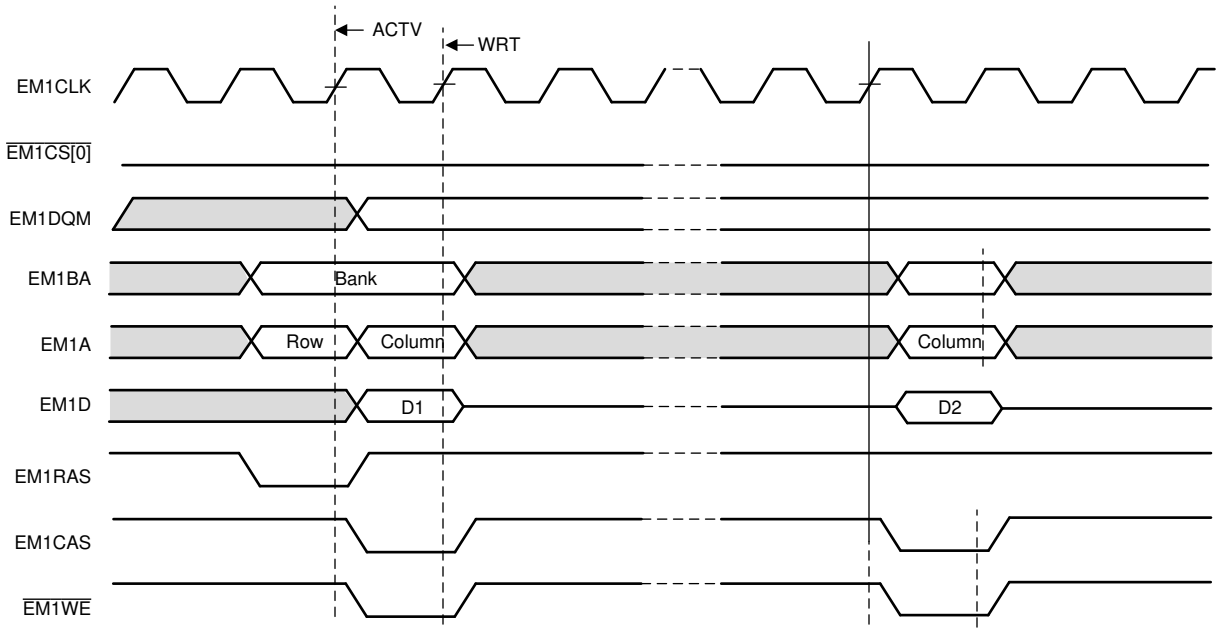


Figure 14-6. Timing Waveform for Basic SDRAM Read Operation

Several other pins are also active during a read access. The EM1DQM[x:0] pins are driven low during the READ commands and are kept low during the NOP commands that correspond to the burst request. The state of the other EMIF pins during each command can be found in Table 14-6.

The EMIF schedules the commands based on the timing information that is provided to the EMIF in the SDRAM timing register (SDRAM_TR). The values for the timing parameters in this register must be chosen to satisfy the timing requirements listed in the SDRAM data manual. The EMIF uses this timing information to avoid violating any timing constraints related to issuing commands. This is commonly accomplished by inserting NOP commands between various commands during an access. Refer to the register description of SDRAM_TR in the SDTIMER register for more details on the various timing parameters.

14.2.5.10 SDRAM Write Operations

When the EMIF receives a write request to SDRAM from one of the requesters listed in Section 14.2.2, the EMIF performs one or more write-access cycles. A write-access cycle begins with the issuing of the ACTV command to select the desired bank and row of the SDRAM device. After the row has been opened, the EMIF proceeds to issue a WRT command while specifying the desired bank and column address. EM1A[10] is held low during the WRT command to avoid auto-precharging. The WRT command signals the SDRAM device to start writing the given data to the specified address while the EMIF issues NOP commands. On this device, burst accesses are not supported; hence, the EMIF issues a WRITE command for each data access. Only when the EMIF SDRAM interface is configured to 16-bit by setting the NM bit of the SDRAM configuration register (SDRAM_CR) to 1 and CPU (or any other controller) does a 32bit WRITE access, a burst access is issued with size of two.

Figure 14-7 shows the signal waveforms for a basic SDRAM write operation.

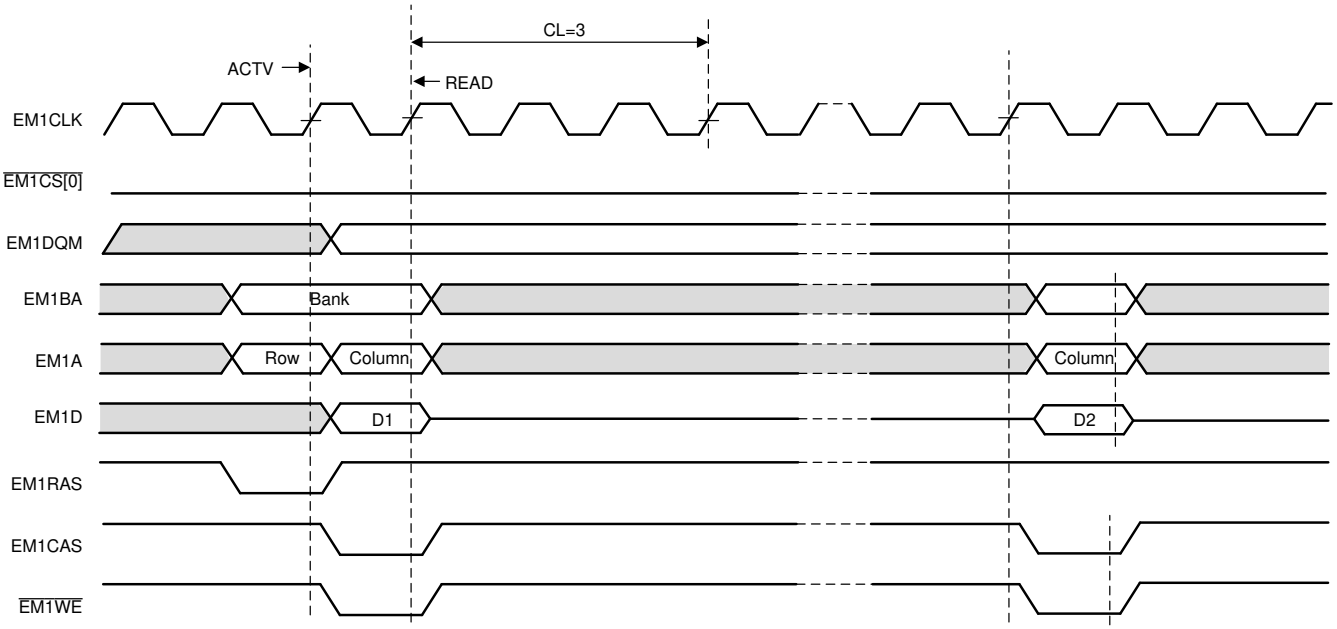


Figure 14-7. Timing Waveform for Basic SDRAM Write Operation

The EMIF truncates a series of bursting data if the remaining addresses of the burst are not part of the write request. The EMIF can truncate the burst in three ways:

- By issuing another WRT to the same page
- By issuing a PRE command to prepare for accessing a different page of the same bank
- By issuing a BT command to prepare for accessing a page in a different bank

Several other pins are also active during a write access. The EM1DQM[x:0] pins are driven to select which bytes of the data word are written to the SDRAM device. The pins are also used to mask out entire undesired data words during a burst access. The state of the other EMIF pins during each command can be found in Table 14-6.

The EMIF schedules the commands based on the timing information that is provided to the EMIF in the SDRAM timing register (SDRAM_TR). The values for the timing parameters in this register must be chosen to satisfy the timing requirements listed in the SDRAM data sheet. The EMIF uses this timing information to avoid violating any timing constraints related to issuing commands. This is commonly accomplished by inserting NOP commands during various cycles of an access. Refer to the register description of SDRAM_TR in the SDTIMR register for more details on the various timing parameters.

14.2.5.11 Mapping from Logical Address to EMIF Pins

When the EMIF receives an SDRAM access request, the EMIF must convert the address of the access into the appropriate signals to send to the SDRAM device. The details of this address mapping are shown in Table 14-14 for 32-bit operation and in Table 14-15 for 16-bit operation. Using the settings of the IBANK and PAGESIZE fields of the SDRAM configuration register (SDRAM_CR), the EMIF determines which bits of the logical address are mapped to the SDRAM row, column, and bank addresses.

As the logical address is incremented by one halfword (16-bit operation), the column address is likewise incremented by one until a page boundary is reached. When the logical address increments across a page boundary, the EMIF moves into the same page in the next bank of the attached device by incrementing the bank address EM1BA and resetting the column address. The page in the previous bank is left open until necessary to close the page. This method of traversal through the SDRAM banks helps maximize the number of open banks inside of the SDRAM and results in an efficient use of the device. There is no limitation on the number of banks that can be open at one time, but only one page within a bank can be open at a time.

The EMIF uses the EM1DQM[3:0] pins during a WRT command to mask out selected bytes or entire words. The EM1DQM[3:0] pins are always low during a READ command.

Table 14-14. Mapping from Logical Address to EMIF Pins for 32-bit SDRAM

IBANK	PAGESIZE	Logical Address													
		31:27	26	25	24	23	22	21:14	13	12	11	10	9	8:1	0
0	0	-						Row Address						Col Address	EM1DQM[2]/EM1DQM[3]
1	0	-						Row Address					EM1BA[0]	Col Address	EM1DQM[2]/EM1DQM[3]
2	0	-						Row Address				EM1BA[1:0]		Col Address	EM1DQM[2]/EM1DQM[3]
0	1	-						Row Address					Column Address		EM1DQM[2]/EM1DQM[3]
1	1	-						Row Address				EM1BA[0]	Column Address		EM1DQM[2]/EM1DQM[3]
2	1	-						Row Address			EM1BA[1:0]		Column Address		EM1DQM[2]/EM1DQM[3]
0	2	-						Row Address				Column Address			EM1DQM[2]/EM1DQM[3]
1	2	-						Row Address			EM1BA[0]	Column Address			EM1DQM[2]/EM1DQM[3]
2	2	-						Row Address		EM1BA[1:0]		Column Address			EM1DQM[2]/EM1DQM[3]
0	3	-						Row Address			Column Address				EM1DQM[2]/EM1DQM[3]
1	3	-						Row Address		EM1BA[0]	Column Address				EM1DQM[2]/EM1DQM[3]
2	3	-						Row Address		EM1BA[1:0]	Column Address				EM1DQM[2]/EM1DQM[3]

Table 14-15. Mapping from Logical Address to EMIF Pins for 16-bit SDRAM

IBANK	PAGESIZE	Logical Address												
		31:26	25	24	23	22	21	20:13	12	11	10	9	8	7:0
0	0	-						Row Address						Col Address
1	0	-						Row Address					EM1BA[0]	Col Address
2	0	-						Row Address				EM1BA[1:0]		Col Address
0	1	-						Row Address					Column Address	
1	1	-						Row Address				EM1BA[0]	Column Address	
2	1	-						Row Address			EM1BA[1:0]		Column Address	
0	2	-						Row Address				Column Address		
1	2	-						Row Address			EM1BA[0]	Column Address		
2	2	-						Row Address		EM1BA[1:0]		Column Address		
0	3	-						Row Address			Column Address			
1	3	-						Row Address		EM1BA[0]	Column Address			
2	3	-						Row Address		EM1BA[1:0]	Column Address			

Note

The upper bit of the row address is used only when addressing 256-Mbit and 512-Mbit SDRAM memories.

14.2.6 Asynchronous Controller and Interface

The EMIF easily interfaces to a variety of asynchronous devices including NOR Flash and SRAM. The EMIF can be operated in two major modes (see Table 14-16):

- Normal Mode
- Select Strobe Mode

Table 14-16. Normal Mode vs. Select Strobe Mode

Mode	Function of EM1DQM pins	Operation of $\overline{\text{EM1CS}}$[4:2]
Normal Mode	Byte enables	Active during the entire asynchronous access cycle
Select Strobe Mode	Byte enables	Active only during the strobe period of an access cycle

The first mode of operation is normal mode, in which the EM1DQM pins of the EMIF function as byte enables. In this mode, the $\overline{\text{EM1CS}}$[4:2] pins behave as typical chip select signals, remaining active for the duration of the asynchronous access. See Section 14.2.6.1 for an example interface with multiple 8-bit devices.

The second mode of operation is select strobe mode, in which the $\overline{\text{EM1CS}}$[4:2] pins act as a strobe, active only during the strobe period of an access. In this mode, the EM1DQM pins of the EMIF function as standard byte enables for reads and writes. A summary of the differences between the two modes of operation are shown in Table 14-16. Refer to Section 14.2.6.4 for the details of asynchronous operations in normal mode, and to Section 14.2.6.5 for the details of asynchronous operations in select strobe mode. The EMIF hardware defaults to normal mode, but can be manually switched to select strobe mode by setting the SS bit in the asynchronous *m* (m = 1, 2, 3, or 4) configuration register (CE*n*CFG) (*n* = 2, 3, or 4). Throughout the chapter, *m* can hold the values 1, 2, 3 or 4; and *n* can hold the values 2, 3, or 4.

The EMIF also provides configurable cycle timing parameters and an extended wait mode that allows the connected device to extend the strobe period of an access cycle. The following sections describe the features related to interfacing with external asynchronous devices.

14.2.6.1 Interfacing to Asynchronous Memory

Figure 14-8 shows the EMIF's external pins used in interfacing with an asynchronous device. In $\overline{\text{EM1CS}}$[n], n = 2, 3, or 4.

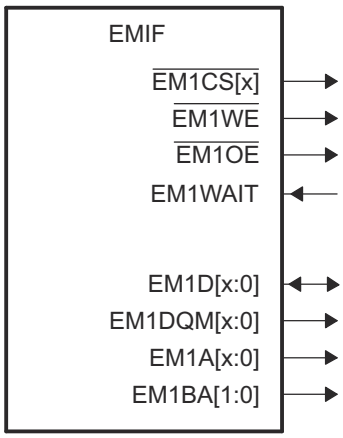
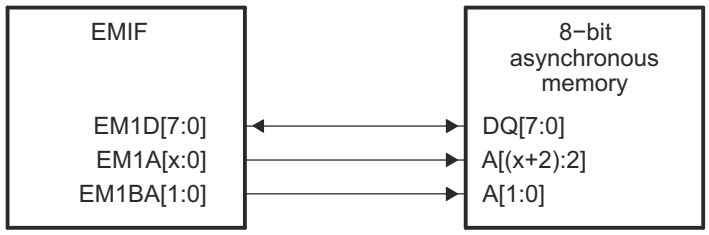


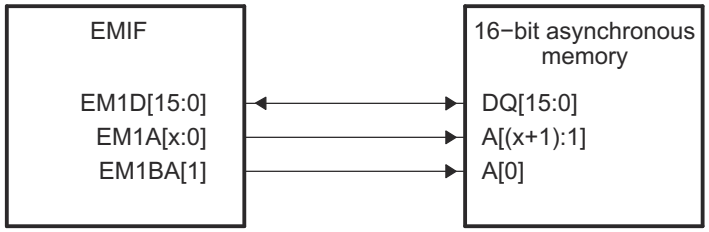
Figure 14-8. EMIF Asynchronous Interface

Of special note is the connection between the EMIF and the external device's address bus. The EMIF address pin EM1A[0] always provides the least-significant bit of a 32-bit word address. Therefore, when interfacing to a 16-bit or 8-bit asynchronous device, the EM1BA[1] and EM1BA[0] pins provide the least-significant bits of the halfword or byte address, respectively. Figure 14-9 and Figure 14-10 show the mapping between the EMIF and the connected device's data and address pins for various programmed data bus widths. The data bus width can be configured in the asynchronous *n* configuration register (ASYNC_CS*n*_CR).

Figure 14-10 shows an interface between the EMIF and an external memory with byte enables. The EMIF must be operated in either normal mode or select strobe mode when using this interface, so that the EM1DQM signals operate as byte enables.



a) EMIF to 8-bit memory interface



b) EMIF to 16-bit memory interface

Figure 14-9. EMIF to 8-bit/16-bit Memory Interface

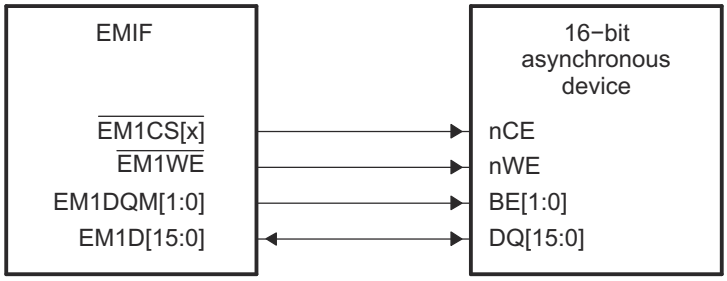


Figure 14-10. Common Asynchronous Interface

14.2.6.2 Accessing Larger Asynchronous Memories

If a device such as a large asynchronous Flash needs to be attached to the EMIF, then GPIO pins can be used to control the Flash device upper address lines.

14.2.6.3 Configuring EMIF for Asynchronous Accesses

The operation of the EMIF's asynchronous interface can be configured by programming the appropriate register fields. The reset value and bit position for each register field can be found in Section 14.6. The following tables list the register fields that can be programmed and describe the purpose of each field. These registers can be programmed prior to accessing the external memory, and the transfer following a write to these registers uses the new configuration.

Table 14-17. Description of the Asynchronous *m* Configuration Register (ASYNC_CSn_CR)

Parameter	Description
SS	**Select Strobe mode.** This bit selects the EMIF's mode of operation in the following way: • SS = 0 selects Normal Mode – EM1DQM pins function as byte enables – $\overline{\text{EM1CS}}$[4:2] active for duration of access • SS = 1 selects Select Strobe Mode – EM1DQM pins function as byte enables – $\overline{\text{EM1CS}}$[4:2] acts as a strobe.
EW	**Extended Wait Mode enable.** • EW = 0 disables extended wait mode • EW = 1 enables extended wait mode When set to 1, the EMIF enables the extended wait mode in which the strobe width of an access cycle can be extended in response to the assertion of the EM1WAIT pin. The WP*n* bit in the asynchronous wait cycle configuration register (ASYNC_WCCR) controls the polarity of the EM1WAIT pin. See Section 14.2.6.6 for more details on this mode of operation.
W_SETUP/R_SETUP	**Read/Write setup widths.** These fields define the number of EMIF clock cycles of setup time for the address pins (EM1A), byte enables (EM1DQM), and asynchronous chip enable ($\overline{\text{EM1CS}}$[4:2]) before the read strobe pin (EM103) or write strobe pin ($\overline{\text{EM1WE}}$) falls, minus one cycle. For writes, the W_SETUP field also defines the setup time for the data pins (EM1D). Refer to the asynchronous device 's data sheet to determine the appropriate setting for this field.
W_STROBE/R_STROBE	**Read/Write strobe widths.** These fields define the number of EMIF clock cycles between the falling and rising of the read strobe pin (EM103) or write strobe pin (EM1WEn), minus one cycle. If Extended Wait Mode is enabled by setting the EW field in the asynchronous *n* configuration register (ASYNC_CS*n*_CR), these fields must be set to a value greater than zero. Refer to the data manual of the external asynchronous device to determine the appropriate setting for this field.
W_HOLD/R_HOLD	**Read/Write hold widths.** These fields define the number of EMIF clock cycles of hold time for the address pins (EM1A and EM1BA), byte enables (EM1DQM), and asynchronous chip enable ($\overline{\text{EM1CS}}$[4:2]) after the read strobe pin (EM103) or write strobe pin ($\overline{\text{EM1WE}}$) rises, minus one cycle. For writes, the W_HOLD field also defines the hold time for the data pins (EM1D). Refer to the data manual of the external asynchronous device to determine the appropriate setting for this field.
TA	**Minimum turnaround time.** This field defines the minimum number of EMIF clock cycles between asynchronous reads and writes, minus one cycle. The purpose of this feature is to avoid contention on the bus. The value written to this field also determines the number of cycles that are inserted between asynchronous accesses and SDRAM accesses. Refer to the data manual of the external asynchronous device to determine the appropriate setting for this field.

Table 14-17. Description of the Asynchronous *m* Configuration Register (ASYNC_CSn_CR) (continued)

Parameter	Description
ASIZE	**Asynchronous Device Bus Width.** This field determines the data bus width of the asynchronous interface in the following way: • ASIZE = 0 selects an 8-bit bus • ASIZE = 1 selects a 16-bit bus • ASIZE = 2 selects a 32-bit bus The configuration of ASIZE determines the function of the EM1A and EM1BA pins as described in Section 14.2.6.1. This field also determines the number of external accesses required to fulfill a request generated by one of the sources mentioned in Section 14.2.2. For example, a request for a 32-bit word requires four external access when ASIZE = 0. Refer to the data manual of the external asynchronous device to determine the appropriate setting for this field.

Table 14-18. Description of the Asynchronous Wait Cycle Configuration Register (ASYNC_WCCR)

Parameter	Description
WP*n*	**EM_WAIT Polarity.** • WP*n* = 0 selects active-low polarity • WP*n* = 1 selects active-high polarity When set to 1, the EMIF waits if the EM1WAIT pin is high. When cleared to 0, the EMIF waits if the EM1WAIT pin is low. The EMIF must have the Extended Wait Mode enabled for the EM1WAIT pin to affect the width of the strobe period.
MAX_EXT_WAIT	**Maximum Extended Wait Cycles.** This field configures the number of EMIF clock cycles the EMIF waits for the EM1WAIT pin to be deactivated during the strobe period of an access cycle. The maximum number of EMIF clock cycles the EMIF waits is determined by the following formula: Maximum Extended Wait Cycles = (MAX_EXT_WAIT + 1) × 16 If the EM1WAIT pin is not deactivated within the time specified by this field, the EMIF resumes the access cycle, registering whatever data is on the bus and proceeding to the hold period of the access cycle. This situation is referred to as an Asynchronous Timeout. An Asynchronous Timeout generates an interrupt, if the interrupt has been enabled in the EMIF interrupt mask set register (INT_MSK_SET). Refer to Section 14.2.9.1 for more information about EMIF interrupts.

Table 14-19. Description of EMIF Interrupt Mask Set Register (INT_MSK_SET)

Parameter	Description
WR_MASK_SET	**Wait Rise Mask Set.** Writing a 1 enables an interrupt to be generated when a rising edge on EM1WAIT occurs.
AT_MASK_SET	**Asynchronous Timeout Mask Set.** Writing a 1 to this bit enables an interrupt to be generated when an Asynchronous Timeout occurs.

Table 14-20. Description of EMIF Interrupt Mast Clear Register (INT_MSK_CLR)

Parameter	Description
WR_MASK_CLR	**Wait Rise Mask Clear.** Writing a 1 to this bit disables the interrupt, clearing the WR_MASK_SET bit in EMIF interrupt mask set register (INT_MSK_SET).
AT_MASK_CLR	**Asynchronous Timeout Mask Clear.** Writing a 1 to this bit prevents an interrupt from being generated when an Asynchronous Timeout occurs.

Note

The EMIF performs SDRAM refreshes even if the SDRAM interface is not used. If using only the ASRAM interface, then SDRAM refreshes impact the ASRAM performance. To avoid this, set PD = 1 in the SDRAM_CR register (Emif1Regs.SDRAM_CR.PD = 1). This bit can be updated only if there are no pending EMIF accesses.

14.2.6.4 Read and Write Operations in Normal Mode

Normal mode is the asynchronous interface default mode of operation. Normal mode is selected when the SS bit in the asynchronous *n* configuration register (ASYNC_CS*n*_CR) is cleared to 0. In this mode, the EM1DQM pins operate as byte enables. Section 14.2.6.4.1 and Section 14.2.6.4.2 explain the details of read and write operations while in normal mode.

14.2.6.4.1 Asynchronous Read Operations (Normal Mode)

Note

During an entire asynchronous read operation, the $\overline{\text{EM1WE}}$ pin is driven high.

An asynchronous read is performed when any of the requesters mentioned in Section 14.2.2 request a read from the attached asynchronous memory. After the request is received, a read operation is initiated once the request becomes the EMIF's highest priority task, according to the priority scheme detailed in Section 14.2.13. In the event that the read request cannot be serviced by a single access cycle to the external device, multiple access cycles are performed by EMIF until the entire request is fulfilled. The details of an asynchronous read operation in normal mode are described in Table 14-21. Also, Figure 14-11 shows an example timing diagram of a basic read operation.

Table 14-21. Asynchronous Read Operation in Normal Mode

Time Interval	Pin Activity in Normal Mode
Turnaround period	Once the read operation becomes the highest priority task for EMIF, the EMIF waits for the programmed number of turn-around cycles before proceeding to the setup period of the operation. The number of wait cycles is taken directly from the TA field of the asynchronous *n* configuration register (ASYNC_CS*n*_CR). Between each access (write or read), the EMIF inserts two cycles of delay even though TA field is programmed as 0. After the EMIF has waited for the turnaround cycles to complete, the EMIF again checks to make sure that the read operation is still the highest priority task. If so, the EMIF proceeds to the setup period of the operation. If the read operation is no longer the highest priority task, the EMIF terminates the operation.
Start of the setup period	The following actions occur at the start of the setup period: • The setup, strobe, and hold values are set according to the R_SETUP, R_STROBE, and R_HOLD values in ASYNC_CS*n*_CR. • The address pins EM1A and EM1BA become valid and carry the values described in Section 14.2.6.1. • $\overline{\text{EM1CS}}$[4:2] falls to enable the external device (if not already low from a previous operation)

Table 14-21. Asynchronous Read Operation in Normal Mode (continued)

Time Interval	Pin Activity in Normal Mode
Strobe period	The following actions occur during the strobe period of a read operation: 1. $\overline{\text{EM1OE}}$ falls at the start of the strobe period 2. On the rising edge of the clock that is concurrent with the end of the strobe period: • $\overline{\text{EM1OE}}$ rises • The data on the EM1Dx bus is sampled by EMIF. In Figure 14-11, EM1WAIT is inactive. If EM1WAIT is instead activated, the strobe period can be extended by the external device to give it more time to provide the data. Section 14.2.6.6 contains more details on using the EM1WAIT pin.
End of the hold period	At the end of the hold period: • The address pins EM1A and EM1BA become invalid • $\overline{\text{EM1CS}}$[4:2] rises (if no more operations are required to complete the current request) The EMIF can be required to issue additional read operations to a device with a small data bus width to complete an entire word access. In this case, the EMIF immediately re-enters the setup period to begin another operation without incurring the turn-round cycle delay. The setup, strobe, and hold values are not updated in this case. If the entire word access has been completed, the EMIF returns to the previous state unless another asynchronous request has been submitted and is currently the highest priority task. If this is the case, EMIF instead enters directly into the turnaround period for the pending read or write operation.

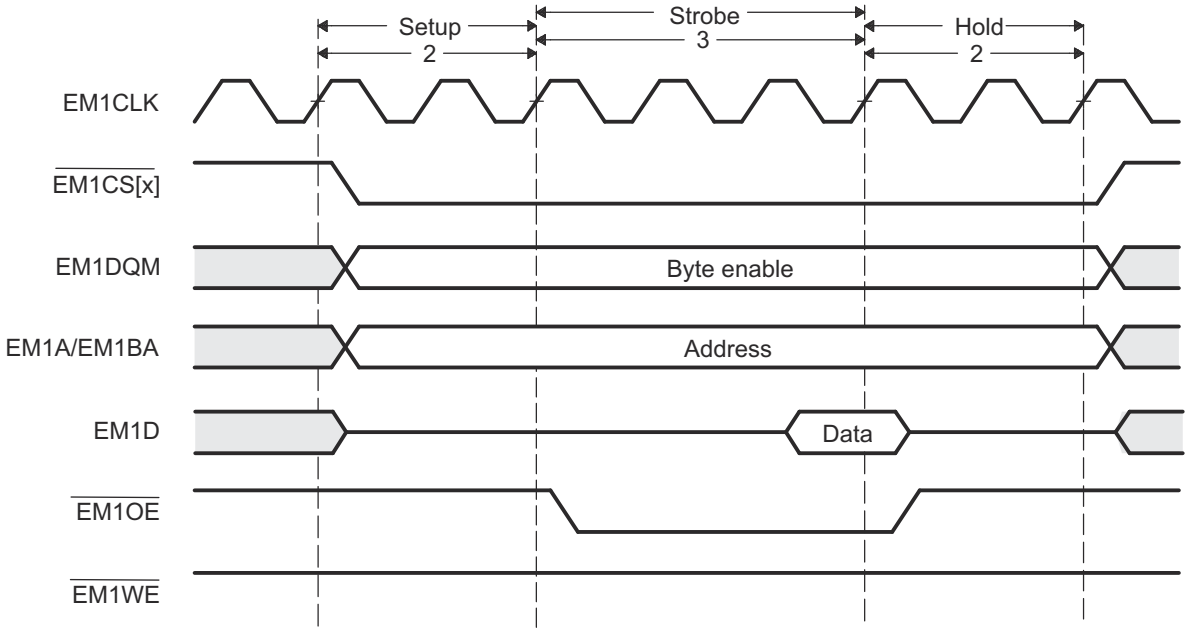


Figure 14-11. Timing Waveform of an Asynchronous Read Cycle in Normal Mode

14.2.6.4.2 Asynchronous Write Operations (Normal Mode)

Note

During an entire asynchronous write operation, the $\overline{\text{EM1OE}}$ pin is driven high.

An asynchronous write is performed when any of the requesters mentioned in Section 14.2.2 request a write to memory in the asynchronous bank of EMIF. After the request is received, a write operation is initiated once the request becomes the EMIF's highest priority task, according to the priority scheme detailed in Section 14.2.13. In the event that the write request cannot be serviced by a single access cycle to the external device, multiple access cycles are performed by the EMIF until the entire request is fulfilled. The details of an asynchronous write operation in normal mode are described in Table 14-22. Also, Figure 14-12 shows an example timing diagram of a basic write operation.

Table 14-22. Asynchronous Write Operation in Normal Mode

Time Interval	Pin Activity in Normal Mode
Turnaround period	Once the write operation becomes the highest priority task for the EMIF, the EMIF waits for the programmed number of turn-around cycles before proceeding to the setup period of the operation. The number of wait cycles is taken directly from the TA field of the asynchronous *n* configuration register (ASYNC_CS*n*_CR). Between each access (write or read) EMIF inserts two cycles of delay even though TA field is programmed as 0. After the EMIF has waited for the turn-around cycles to complete, the EMIF again checks to make sure that the write operation is still the highest priority task. If so, the EMIF proceeds to the setup period of the operation. If the write operation is no longer the highest priority task, EMIF terminates the operation.
Start of the setup period	The following actions occur at the start of the setup period: • The setup, strobe, and hold values are set according to the W_SETUP, W_STROBE, and W_HOLD values in ASYNC_CS*n*_CR. • The address pins EM1A and EM1BA and the data pins EM1Dx become valid. The EM1A and EM1BA pins carry the values described in Section 14.2.6.1. • $\overline{\text{EM1CS}}$[4:2] falls to enable the external device (if not already low from a previous operation).
Strobe period	The following actions occur at the start of the strobe period of a write operation: 1. $\overline{\text{EM1WE}}$ falls 2. The EM1DQM pins become valid as byte enables. The following actions occur on the rising edge of the clock that is concurrent with the end of the strobe period: 1. $\overline{\text{EM1WE}}$ rises 2. The EM1DQM pins deactivate In Figure 14-12, EM1WAIT is inactive. If EM1WAIT is instead activated, the strobe period can be extended by the external device to give it more time to accept the data. Section 14.2.6.6 contains more details on using the EM1WAIT pin.
End of the hold period	At the end of the hold period: • The address pins EM1Ax and EM1BAx become invalid • The data pins become invalid • $\overline{\text{EM1CS}}$[n] (n = 2, 3, or 4) rises (if no more operations are required to complete the current request) The EMIF can be required to issue additional write operations to a device with a small data bus width to complete an entire word access. In this case, the EMIF immediately re-enters the setup period to begin another operation without incurring the turnaround cycle delay. The setup, strobe, and hold values are not updated in this case. If the entire word access has been completed, the EMIF returns to the previous state unless another asynchronous request has been submitted and is currently the highest priority task. If this is the case, the EMIF instead enters directly into the turnaround period for the pending read or write operation.

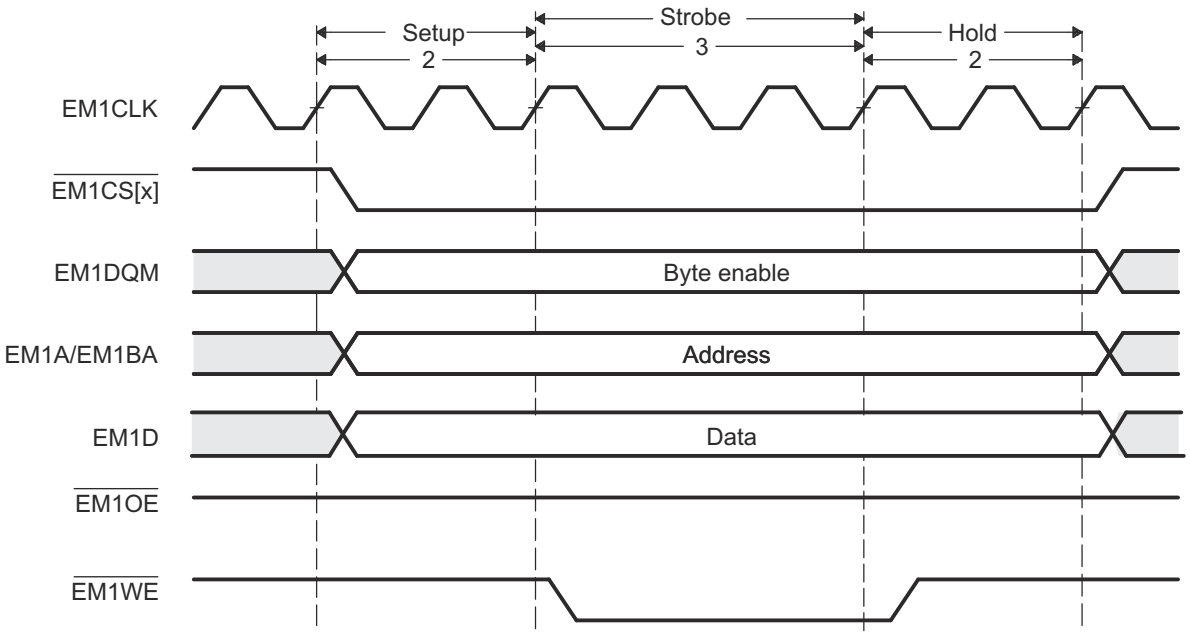


Figure 14-12. Timing Waveform of an Asynchronous Write Cycle in Normal Mode

14.2.6.5 Read and Write Operation in Select Strobe Mode

Select Strobe mode is the EMIF's second mode of operation. Select Strobe mode is selected when the SS bit of the asynchronous *n* configuration register (ASYNC_CSn_CR) is set to 1. In this mode, the EM1DQM pins operate as byte enables and the $\overline{\text{EM1CS}}$[n] (n = 2, 3, or 4) pin is only active during the strobe period of an access cycle. Section 14.2.6.4.1 and Section 14.2.6.4.2 explain the details of read and write operations while in select strobe mode.

14.2.6.5.1 Asynchronous Read Operations (Select Strobe Mode)

Note

During the entirety of an asynchronous read operation, the EM1WEn pin is driven high.

An asynchronous read is performed when any of the requesters mentioned in Section 14.2.2 request a read from the attached asynchronous memory. After the request is received, a read operation is initiated once the request becomes the EMIF's highest priority task, according to the priority scheme detailed in Section 14.2.13. In the event that the read request cannot be serviced by a single access cycle to the external device, multiple access cycles are performed by the EMIF until the entire request is fulfilled. The details of an asynchronous read operation in select strobe mode are described in Table 14-23. Also, Figure 14-13 shows an example timing diagram of a basic read operation.

Table 14-23. Asynchronous Read Operation in Select Strobe Mode

Time Interval	Pin Activity in Select Strobe Mode
Turnaround period	Once the read operation becomes the highest priority task for the EMIF, the EMIF waits for the programmed number of turnaround cycles before proceeding to the setup period of the operation. The number of wait cycles is taken directly from the TA field of the asynchronous *n* configuration register (ASYNC_CSn_CR). Between each access (Write or Read) EMIF inserts two cycles of delay even though TA field is programmed as 0. After the EMIF has waited for the turn-around cycles to complete, the EMIF again checks to make sure that the read operation is still the highest priority task. If so, the EMIF proceeds to the setup period of the operation. If the read operation is no longer the highest priority task, the EMIF terminates the operation.
Start of the setup period	The following actions occur at the start of the setup period: • The setup, strobe, and hold values are set according to the R_SETUP, R_STROBE, and R_HOLD values in ASYNC_CS*n*_CR. • The address pins EM1A and EM1BA become valid and carry the values described in Section 14.2.6.1. • The EM1DQM pins become valid as byte enables.

Strobe period	The following actions occur during the strobe period of a read operation:
	1. $\overline{\text{EM1CS}}$[n] (n = 2, 3, or 4) and $\overline{\text{EM1OE}}$ fall at the start of the strobe period
	2. On the rising edge of the clock that is concurrent with the end of the strobe period:
	  • $\overline{\text{EM1CS}}$[n] (n = 2, 3, or 4) and $\overline{\text{EM10E}}$ rise
	  • The data on the EM1D bus is sampled by EMIF.
	In Figure 14-13, EM1WAIT is inactive. If EM1WAIT is instead activated, the strobe period can be extended by the external device to give more time to provide the data. Section 14.2.6.6 contains more details on using the EM1WAIT pin.

End of the hold period	At the end of the hold period: • The address pins EM1A and EM1BA become invalid • The EM1DQM pins become invalid The EMIF can be required to issue additional read operations to a device with a small data bus width to complete an entire word access. In this case, the EMIF immediately re-enters the setup period to begin another operation without incurring the turnaround cycle delay. The setup, strobe, and hold values are not updated in this case. If the entire word access has been completed, the EMIF returns to the previous state unless another asynchronous request has been submitted and is currently the highest priority task. If this is the case, the EMIF instead enters directly into the turnaround period for the pending read or write operation.

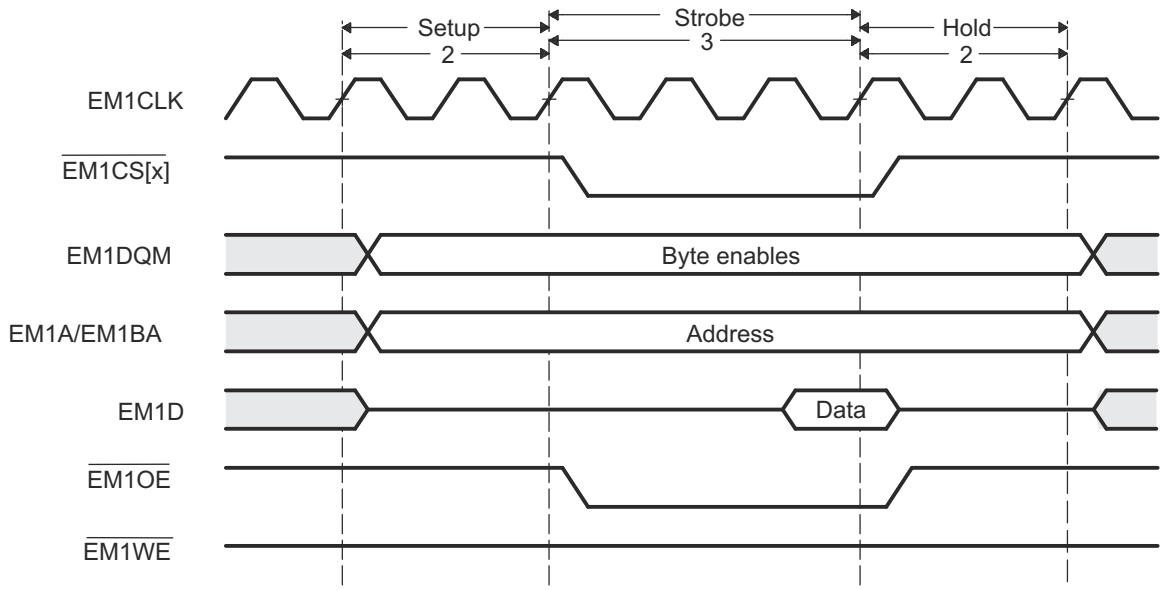


Figure 14-13. Timing Waveform of an Asynchronous Read Cycle in Select Strobe Mode

14.2.6.5.2 Asynchronous Write Operations (Select Strobe Mode)

Note

During the entirety of an asynchronous write operation, the $\overline{\text{EM1OE}}$ pin is driven high.

An asynchronous write is performed when any of the requesters mentioned in Section 14.2.2 request a write to memory in the asynchronous bank of EMIF. After the request is received, a write operation is initiated once the request becomes the EMIF's highest priority task, according to the priority scheme detailed in Section 14.2.13. In the event that the write request cannot be serviced by a single access cycle to the external device, multiple access cycles are performed by the EMIF until the entire request is fulfilled. The details of an asynchronous write operation in select strobe mode are described in Table 14-24. Also, Figure 14-14 shows an example timing diagram of a basic write operation.

Table 14-24. Asynchronous Write Operation in Select Strobe Mode

Time Interval	Pin Activity in Select Strobe Mode
Turnaround period	Once the write operation becomes the highest priority task for the EMIF, the EMIF waits for the programmed number of turnaround cycles before proceeding to the setup period of the operation. The number of wait cycles is taken directly from the TA field of the asynchronous *n* configuration register (ASYNC_CSn_CR). Between each access (Write or Read) EMIF inserts two cycles of delay even though TA field is programmed as 0. After the EMIF has waited for the turnaround cycles to complete, the EMIF again checks to make sure that the write operation is still the highest priority task. If so, the EMIF proceeds to the setup period of the operation. If the write operation is no longer the highest priority task, the EMIF terminates the operation.
Start of the setup period	The following actions occur at the start of the setup period: • The setup, strobe, and hold values are set according to the W_SETUP, W_STROBE, and W_HOLD values in ASYNC_CSn_CR. • The address pins EM1A and EM1BA and the data pins EM1D become valid. The EM1A and EM1BA pins carry the values described in Section 14.2.6.1. • The EM1DQM pins become active as byte enables.
Strobe period	The following actions occur at the start of the strobe period of a write operation: • $\overline{\text{EM1CS}}$[n] (n = 2, 3, or 4) and $\overline{\text{EM1WE}}$ fall The following actions occur on the rising edge of the clock which is concurrent with the end of the strobe period: • $\overline{\text{EM1CS}}$[n] (n = 2, 3, or 4) and $\overline{\text{EM1WE}}$ rise In Figure 14-14, EM1WAIT is inactive. If EM1WAIT is instead activated, the strobe period can be extended by the external device to give more time to accept the data. Section 14.2.6.6 contains more details on using the EM1WAIT pin.
End of the hold period	At the end of the hold period: • The address pins EM1A and EM1BA become invalid • The data pins become invalid • The EM1DQM pins become invalid The EMIF can be required to issue additional write operations to a device with a small data bus width to complete an entire word access. In this case, the EMIF immediately re-enters the setup period to begin another operation without incurring the turnaround cycle delay. The setup, strobe, and hold values are not updated in this case. If the entire word access has been completed, the EMIF returns to the previous state unless another asynchronous request has been submitted and is currently the highest priority task. If this is the case, the EMIF instead enters directly into the turnaround period for the pending read or write operation.

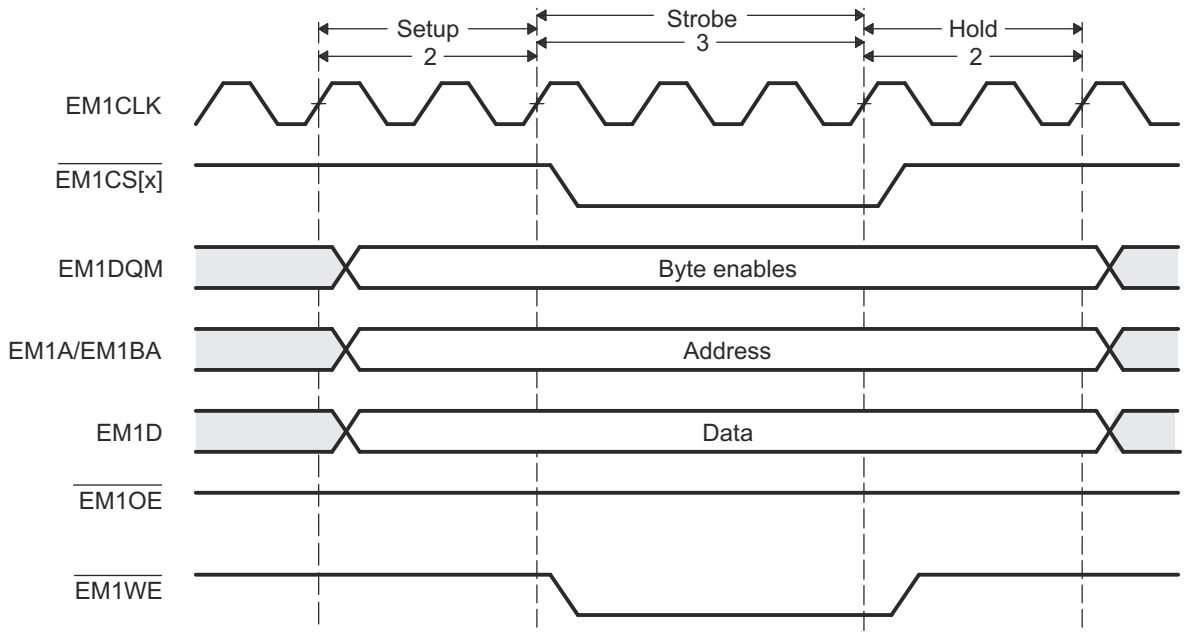


Figure 14-14. Timing Waveform of an Asynchronous Write Cycle in Select Strobe Mode

14.2.6.6 Extended Wait Mode and the EM1WAIT Pin

The EMIF supports the extended wait mode. This is a mode that the external asynchronous device can assert control over the length of the strobe period. The extended wait mode can be entered by setting the EW bit in the asynchronous *n* configuration register (ASYNC_CS*n*_CR (*n* = 2, 3, or 4). When this bit is set, the EMIF monitors the EM1WAIT pin to determine if the attached device wishes to extend the strobe period of the current access cycle beyond the programmed number of clock cycles.

When the EMIF detects that the EM1WAIT pin has been asserted, the EMIF begins inserting extra strobe cycles into the operation until the EM1WAIT pin is deactivated by the external device. The EMIF then returns to the last cycle of the programmed strobe period and the operation proceeds as usual from this point. Refer to the device data sheet for details on the timing requirements of the EM1WAIT signal.

The EM1WAIT pin cannot be used to extend the strobe period indefinitely. The programmable MAX_EXT_WAIT field in the asynchronous wait cycle configuration register (AWCC) determines the maximum number of EM1CLK cycles the strobe period can be extended beyond the programmed length. When the counter expires, the EMIF proceeds to the hold period of the operation regardless of the state of the EM1WAIT pin. The EMIF can also generate an interrupt upon expiration of this counter. See Section 14.2.9.1 for details on enabling this interrupt.

For the EMIF to function properly in the extended wait mode, the WP*n* bit of AWCC must be programmed to match the polarity of the EM1WAIT pin. In the reset state of 1, the EMIF inserts wait cycles when the EM1WAIT pin is sampled high. When set to 0, the EMIF inserts wait cycles only when EM1WAIT is sampled low. This programmability allows for a glueless connection to larger variety of asynchronous devices.

Finally, a restriction is placed on the strobe period timing parameters when operating in extended wait mode. Specifically, the sum of the W_SETUP and W_STROBE fields must be greater than four, and the sum of the R_SETUP and R_STROBE fields must be greater than four for the EMIF to recognize the EM1WAIT pin has been asserted. The W_SETUP, W_STROBE, R_SETUP, and R_STROBE fields are in ASYNC_CS*n*_CR.

14.2.7 Data Bus Parking

The EMIF always drives the data bus to the previous write data value when the EMIF is idle. This feature is called data bus parking. Only when the EMIF issues a read command to the external memory does the EMIF stop driving the data bus. After the EMIF latches the last read data, the EMIF immediately parks the data bus again.

The one exception to this behavior occurs after performing an asynchronous read operation while the EMIF is in the self-refresh state. In this situation, the read operation is not followed by the EMIF parking the data bus. Instead, the EMIF tri-states the data bus. Therefore, it is not recommended to perform asynchronous read operations while the EMIF is in the self-refresh state, to prevent floating inputs on the data bus. External pull-ups, such as 10-kohm resistors, must be placed on the 16 EMIF data bus pins (that do not have internal pull-ups) if required to perform reads in this situation. The precise resistor value must be chosen so that the worst case combined off-state leakage currents do not cause the voltage levels on the associated pins to drop below the high-level input voltage requirement.

For information about the self-refresh state, see Section 14.2.5.7.

14.2.8 Reset and Initialization Considerations

The EMIF memory controller has two active-low reset signals, CHIP_RST_n and MOD_G_RST_n. Both these reset signals are driven by the device system reset signal. This device does not offer the flexibility to reset just the EMIF state machine without also resetting the EMIF controller's memory-mapped registers. As soon as the device system reset is released (driven high), the EMIF memory controller immediately begins its initialization sequence. Command and data stored in the EMIF memory controller FIFOs are lost. Refer to Section 14.2 for more information on conditions that can cause a device system reset to be asserted.

When system reset is released, the EMIF automatically begins running the SDRAM initialization sequence described in Section 14.2.5.4. Even though the initialization procedure is automatic, a special procedure, found in Section 14.2.5.5 must still be followed.

14.2.9 Interrupt Support

The EMIF supports a single interrupt to the CPU. Section 14.2.9.1 details the generation and internal masking of EMIF interrupts.

14.2.9.1 Interrupt Events

There are three conditions that can cause the EMIF to generate an interrupt to the CPU. These conditions are:

- A rising edge on the EM1WAIT signal (wait rise interrupt)
- An asynchronous time out
- Usage of unsupported addressing mode (line trap interrupt)

The wait rise interrupt occurs when a rising edge is detected on EM1WAIT signal. This interrupt generation is not affected by the WP*n* bit in the asynchronous wait cycle configuration register (ASYNC_WCCR). The asynchronous time out interrupt condition occurs when the attached asynchronous device fails to deassert the EM1WAIT pin within the number of cycles defined by the MAX_EXT_WAIT bit in AWCC (this happens only in extended wait mode). The EMIF supports only linear incrementing and cache line wrap addressing modes . If an access request for an unsupported addressing mode is received, the EMIF sets the LT bit in the EMIF interrupt raw register (INT_RAW) and treats the request as a linear incrementing request.

Only when the interrupt is enabled by setting the appropriate bit (WR_MASK_SET/AT_MASK_SET/ LT_MASK_SET) in the EMIF interrupt mask set register (INT_MSK_SET) to 1, is the interrupt sent to the CPU. Once enabled, the interrupt can be disabled by writing a 1 to the corresponding bit in the EMIF interrupt mask clear register (INT_MSK_CLR). The bit fields in both the INT_MSK_SET and INT_MSK_CLR can be used to indicate whether the interrupt is enabled. When the interrupt is enabled, the corresponding bit field in both the INT_MSK_SET and INT_MSK_CLR have a value of 1; when the interrupt is disabled, the corresponding bit field has a value of 0.

The EMIF interrupt raw register (INT_RAW) and the IF interrupt mask register (INT_MSK) indicate the status of each interrupt. The appropriate bit (WR/AT/LT) in INT_RAW is set when the interrupt condition occurs, whether or not the interrupt has been enabled. However, the appropriate bit (WR_MASKED/AT_MASKED/LT_MASKED) in INT_MSK is set only when the interrupt condition occurs and the interrupt is enabled. Writing a 1 to the bit in INT_RAW clears the INT_RAW bit as well as the corresponding bit in INT_MSK. Table 14-25 contains a brief summary of the interrupt status and control bit fields. See Section 14.6 for complete details on the register fields.

Table 14-25. Interrupt Monitor and Control Bit Fields

Register Name	Bit Name	Description
EMIF interrupt raw register (INT_RAW)	WR	This bit is set when an rising edge on the EM1WAIT signal occurs. Writing a 1 clears the WR bit as well as the WR_MASKED bit in INT_MSK.
	AT	This bit is set when an asynchronous timeout occurs. Writing a 1 clears the AT bit as well as the AT_MASKED bit in INT_MSK.
	LT	This bit is set when an unsupported addressing mode is used. Writing a 1 clears LT bit as well as the LT_MASKED bit in INT_MSK.
EMIF interrupt mask register (INT_MSK)	WR_MASKED	This bit is set only when a rising edge on the EM1WAIT signal occurs and the interrupt has been enabled by writing a 1 to the WR_MASK_SET bit in INT_MSK_SET.
	AT_MASKED	This bit is set only when an asynchronous timeout occurs and the interrupt has been enabled by writing a 1 to the AT_MASK_SET bit in INT_MSK_SET.
	LT_MASKED	This bit is set only when line trap interrupt occurs and the interrupt has been enabled by writing a 1 to the LT_MASK_SET bit in INT_MSK_SET.
EMIF interrupt mask set register (INT_MSK_SET)	WR_MASK_SET	Writing a 1 to this bit enables the wait rise interrupt.
	AT_MASK_SET	Writing a 1 to this bit enables the asynchronous timeout interrupt.
	LT_MASK_SET	Writing a 1 to this bit enables the line trap interrupt.
EMIF interrupt mask clear register (INT_MSK_CLR)	WR_MASK_CLR	Writing a 1 to this bit disables the wait rise interrupt.
	AT_MASK_CLR	Writing a 1 to this bit disables the asynchronous timeout interrupt.
	LT_MASK_CLR	Writing a 1 to this bit disables the line trap interrupt.

14.2.10 RTDMA Event Support

The EMIF memory controller is a RTDMA target peripheral and therefore does not generate RTDMA events. Data read and write requests can be made directly, by controllers and the RTDMA.

14.2.11 EMIF Signal Multiplexing

For details on the EMIF signal multiplexing, see the *GPIO* chapter, I/O Multiplexing Module section, of this technical reference manual.

14.2.12 Memory Map

For information describing the device memory-map, see the data sheet.

14.2.13 Priority and Arbitration

Section 14.2.2 describes the external prioritization and arbitration among requests from different sources within the microcontroller. The result of this external arbitration is that only one request is presented to the EMIF at a time. Once the EMIF completes a request, the external arbiter then provides the EMIF with the next pending request.

Internally, the EMIF undertakes memory device transactions according to a strict priority scheme. The highest priority events are:

- A device reset.
- A write to any of the three least significant bytes of the SDRAM configuration register (SDRAM_CR).

Either of these events causes the EMIF to immediately commence the initialization sequence as described in Section 14.2.5.4.

Once the EMIF has completed its initialization sequence, the EMIF performs memory transactions according to the following priority scheme (highest priority listed first):

1. If the EMIF's backlog refresh counter is at the Refresh Must urgency level, the EMIF performs multiple SDRAM auto-refresh cycles until the Refresh Release urgency level is reached.
2. If an SDRAM or asynchronous read has been requested, the EMIF performs a read operation.
3. If the EMIF's backlog refresh counter is at the Refresh Need urgency level, the EMIF performs an SDRAM auto-refresh cycle.
4. If an SDRAM or asynchronous write has been requested, the EMIF performs a write operation.
5. If the EMIF's backlog refresh counter is at the Refresh May or Refresh Release urgency level, the EMIF performs an SDRAM auto-refresh cycle.
6. If the value of the SR bit in SDRAM_CR has been set to 1, the EMIF enters the self-refresh state as described in Section 14.2.5.7.

After taking one of the actions listed above, the EMIF then returns to the top of the priority list to determine the next action.

Because the EMIF does not issue auto-refresh cycles when in the self-refresh state, the above priority scheme does not apply when in this state. See Section 14.2.5.7 for details on the operation of the EMIF when in the self-refresh state.

14.2.14 System Considerations

This section describes various system considerations to keep in mind when operating the EMIF.

14.2.14.1 Asynchronous Request Times

In a system that interfaces to both SDRAM and asynchronous memory, the asynchronous requests must not take longer than the smaller of the following two values:

- t_{RAS} (typically 120µs) - to avoid violating the maximum time allowed between issuing an ACTV and PRE command to the SDRAM.
- $t_{Refresh\ Rate}$ × 11 (typically 15.7µs × 11 = 172.7µs) - to avoid refresh violations on the SDRAM.

The length of an asynchronous request is controlled by multiple factors, the primary factor being the number of access cycles required to complete the request. For example, an asynchronous request for 4 bytes requires four access cycles using an 8-bit data bus and only two access cycle using a 16-bit data bus. The maximum request size that the EMIF can be sent is 16 words; therefore, the maximum number of access cycles per memory request is 64 when the EMIF is configured with an 8-bit data bus. The length of the individual access cycles that make up the asynchronous request is determined by the programmed setup, strobe, hold, and turnaround values, but can also be extended with the assertion of the EM1WAIT input signal up to a programmed maximum limit. The user must make sure that an entire asynchronous request does not exceed the timing values listed above when also interfacing to an SDRAM device. This can be done by limiting the asynchronous timing parameters.

14.2.15 Power Management

Power dissipation from the EMIF memory controller can be managed by following methods:

- Self-refresh mode
- Power-down mode
- Gating input clocks to the module off

Gating input clocks off to the EMIF memory controller achieves higher power savings when compared to the power savings of self-refresh or power down mode. The input clock to EMIF can be turned off through the use of the Global Clock Module (GCM). Before gating clocks off, the EMIF memory controller must place the SDR SDRAM memory in self-refresh mode. If the external memory requires a continuous clock, the VCLK3 clock domain must not be turned off because this can result in data corruption. See the following subsections for the proper procedures to follow when stopping EMIF memory controller clocks.

14.2.15.1 Power Management Using Self-Refresh Mode

The EMIF memory controller can be placed into a self-refresh state in order to place the attached SDRAM devices into self-refresh mode, which consumes less power for most SDRAM devices. In this state, the attached SDRAM device uses an internal clock to perform its own auto refresh cycles. This maintains the validity of the data in the SDRAM without the need for any external commands. Refer to Section 14.2.5.7 for more details on placing the EMIF into the self-refresh state.

14.2.15.2 Power Management Using Power Down Mode

In the power down mode, the EMIF drives EM1SDCKE low to lower the power consumption. EM1SDCKE goes high when there is a need to send refresh (REFR) commands, after which EM1SDCKE is again driven low. EM1SDCKE remains low until any request arrives. Refer to Section 14.2.5.8 for more details on placing the EMIF in power-down mode.

14.2.16 Emulation Considerations

The EMIF remains fully functional during emulation halts in order to allow emulation access to external memory.

14.3 EMIF Subsystem (EMIFSS)

This section details the configuration of the EMIF subsystem, which is used by memory access initiators to support access to the EMIF peripheral.

Figure 14-15 shows that different initiators are connected to the fast and slow access ports. Fast access ports are limited to the primary CPU. Slow access ports take an additional cycle for accesses, but have mechanisms to improve throughput for certain scenarios like RTDMA accesses. Debug accesses are connected to slow access ports and are arbitrated externally from the EMIFSS. The round-robin arbitration scheme is used among all fast and slow access ports as follows:

1. CPU1 (default)
2. CPU2
3. CPU3
4. RTDMA1
5. RTDMA2
6. Debug Bridge

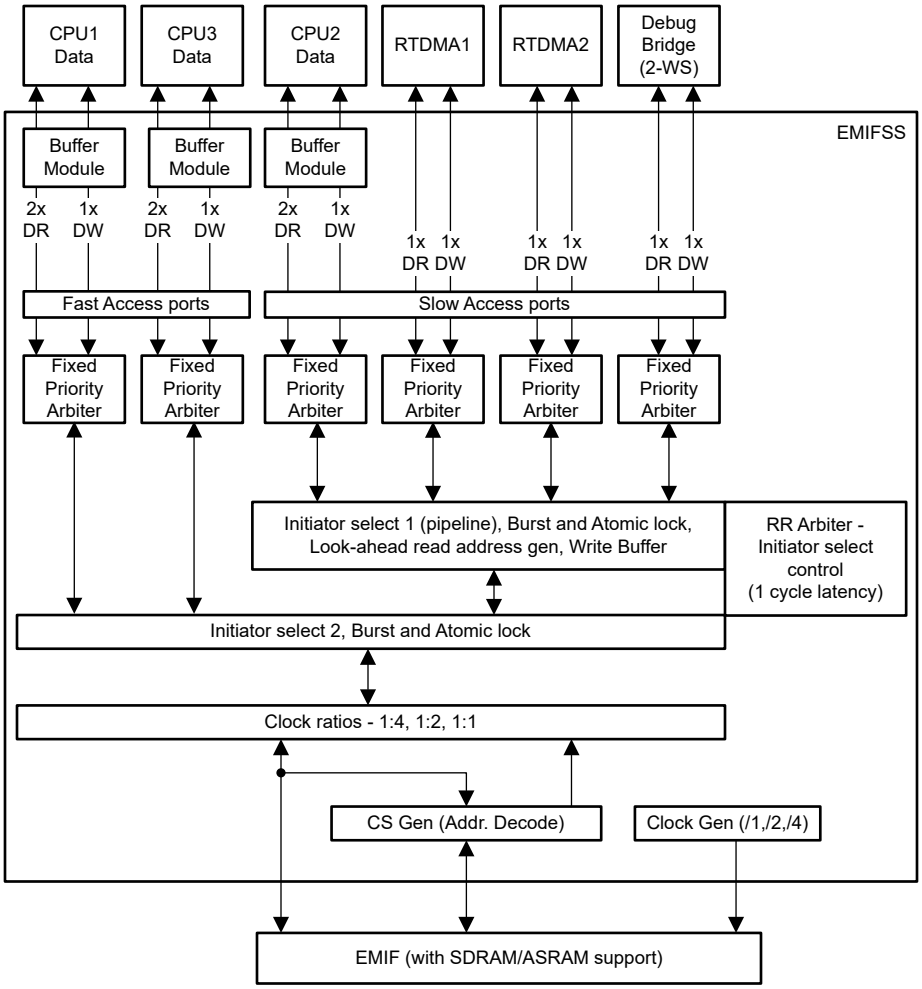


Figure 14-15. EMIFSS Block Diagram

14.3.1 Burst Support

The RTDMA is connected through a slow access port, so all accesses take an extra cycle as opposed to the fast access port. For this reason, the RTDMA is enhanced to support local address generation within the EMIFSS arbiter initiator select 1 component. This enables read burst performance close to that of the fast access port, thereby improving throughput. For more information, refer to the *Real-Time Direct Memory Access (RTDMA)* chapter.

14.3.2 EMIFSS Performance Improvement

The C29x EMIFSS enables devices to improve access performance using the inherent burst support and dataline buffer for each CPU. The following tables show the improvement increase for different accesses.

Table 14-26. SDRAM Sequential Bulk Transfer

Access Type[1]	Access Time on C28x (Cycles)	Access Time on C29x (Cycles)	C28x:C29x Improvement	Access Time with DMA (Cycles)	Access Time with RTDMA (Cycles)	DMA:RTDMA Improvement
SDRAM to SRAM 16-bit Access	8196	2821[2]	2.9X	9267	Not Available[3]	-
SDRAM to SRAM 32-bit Access	5125	2564[2]	2.0X	4633	1125	4.1X
SDRAM to SRAM 64-bit Access	Not Available	2559	-	Not Available	1125	-

Table 14-26. SDRAM Sequential Bulk Transfer (continued)

Access Type[1]	Access Time on C28x (Cycles)	Access Time on C29x (Cycles)	C28x:C29x Improvement	Access Time with DMA (Cycles)	Access Time with RTDMA (Cycles)	DMA:RTDMA Improvement
SRAM to SRDAM 16-bit Access	4097	4100[2]	1.0X	5172	Not Available[3]	-
SRAM to SDRAM 32-bit Access	3076	2048[2]	1.5X	2564	1089	2.4X
SRAM to SDRAM 64-bit Access	Not Available	1535	-	Not Available	1093	-

(1) There can be a difference in timings due to the refresh cycles incurred by the SDRAM
(2) C29x numbers shown for 16-bit and 32-bit accesses are with the read dataline buffer enabled
(3) RTDMA accesses are not supported for 16-bit transfers in a burst

Table 14-27. SDRAM Random Access Performance

Access Type	New Row Access (SYSCLK Cycles)	Opened Row Access (SYSCLK Cycles)
8-Bit Read	20	16
8-Bit Write	12	8
16-Bit Read	20	16
16-Bit Write	12	8
32-Bit Read	20	16
32-Bit Write	12	8
64-Bit Read	24	20
64-Bit Write	16	12

Table 14-28. ASRAM Sequential Bulk Transfer (1:1:1 Mode)

Access Type	Access Time on C28x (Cycles)	Access Time on C29x (Cycles)	C28x:C29x Improvement	Access Time with DMA (Cycles)	Access Time with RTDMA (Cycles)	DMA:RTDMA Improvement
ASRAM to SRAM 16-bit Access	2560	2176[1]	1.2X	4095	Not Available[2]	-
ASRAM to SRAM 32-bit Access	3330	1793[1]	1.9X	2815	1553	1.8X
ASRAM to SRAM 64-bit Access	Not Available	1792	-	Not Available	1553	-
SRAM to ASRAM 16-bit Access	2561	2559[1]	1.0X	3583	Not Available[2]	-
SRAM to ASRAM 32-bit Access	3327	2047[1]	1.6X	2559	1598	1.6X
SRAM to ASRAM 64-bit Access	Not Available	1793	-	Not Available	1598	-

(1) C29x numbers shown for 16-bit and 32-bit accesses are with the read dataline buffer enabled
(2) RTDMA accesses are not supported for 16-bit transfers in a burst

Table 14-29. ASRAM Sequential Bulk Transfer (1:4:1 Mode)

Access Type	Access Time on C28x (Cycles)	Access Time on C29x (Cycles)	C28x:C29x Improvement	Access Time with DMA (Cycles)	Access Time with RTDMA (Cycles)	DMA:RTDMA Improvement
ASRAM to SRAM 16-bit Access	4096	3712(1)	1.1X	5631	Not Available(2)	-
ASRAM to SRAM 32-bit Access	4867	3329(1)	1.5X	4351	3089	1.4X
ASRAM to SRAM 64-bit Access	Not Available	3328	-	Not Available	3089	-
SRAM to ASRAM 16-bit Access	4097	4095(1)	1.0X	5119	Not Available(2)	-
SRAM to ASRAM 32-bit Access	4863	3583(1)	1.4X	4095	3134	1.3X
SRAM to ASRAM 64-bit Access	Not Available	3329	-	Not Available	3134	-

(1) C29x numbers shown for 16-bit and 32-bit accesses are with the read dataline buffer enabled
(2) RTDMA accesses are not supported for 16-bit transfers in a burst

Table 14-30. ASRAM Random Access Performance

Access Type	1:1:1 Configuration (SYSCLK Cycles)	1:4:1 Configuration (SYSCLK Cycles)
8-Bit Read	5	8
8-Bit Write	4	7
16-Bit Read	5	8
16-Bit Write	4	7
32-Bit Read	8	14
32-Bit Write	7	13
64-Bit Read	14	26
64-Bit Write	13	25

14.3.3 Buffer Module

The buffer module is implemented for each CPU and consists of the following:

- A dataline buffer that services the data read 1/2 accesses (DR1/DR2)
- A write FIFO that services the data write accesses (DW)
- Associated logic to conditionally bypass the dataline buffer

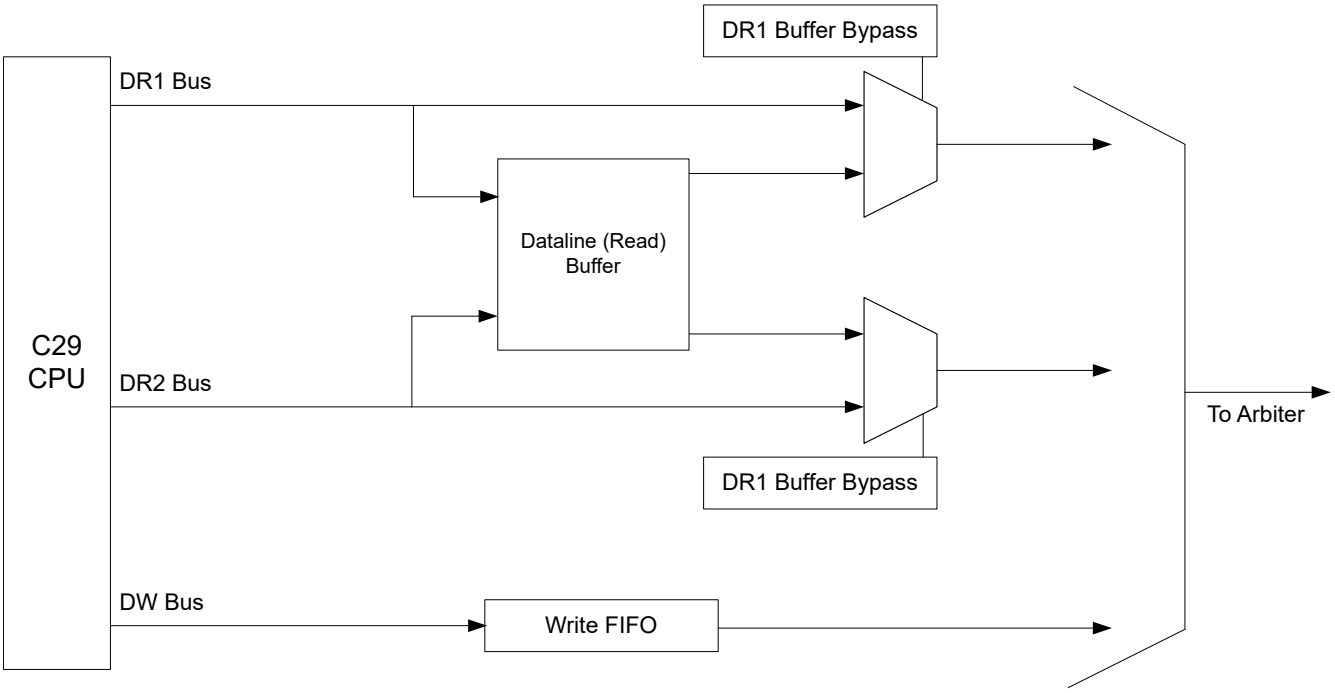


Figure 14-16. Buffer Module Block Diagram

Any 16-bit or 32-bit read which misses the dataline buffer initiates a 64-bit access to the EMIF and stalls the CPU until the four 16-bit access completes and refills the buffer (on a 16-bit SDRAM or ASRAM). This means that sequential accesses have better performance.

If the accesses are not sequential memory locations, dual-mapped regions are provided for the software to chosse between the following:

- The Primary Memory Map - This uses the dataline buffer for reads and is optimized for sequential accesses
- The Secondary Memory Map - This bypasses the dataline buffer for reads and is optimized for random accesses

For DR1 and DR2 accesses, the EMIF SDRAM and ASRAM memory regions for each of the CSx are dual-mapped. The EMIF register regions are not dual-mapped and always bypass the buffer. DW accesses to either of the dual memory maps behave identically and always go through the write FIFO.

Table 14-31 is an example of the memory map for EMIF. For actual address range information, see the device data sheet memory map.

Table 14-31. Dual Memory Map Example

Address Range	EMIF Region
A	EMIF SDRAM CS0 (Primary Map)
B	EMIF ASRAM CS2 (Primary Map)
C	EMIF ASRAM CS3 (Primary Map)
E	EMIF ASRAM CS4 (Primary Map)
E	EMIF Configuration Registers
F	EMIF SDRAM CS0 (Secondary Map)
G	EMIF ASRAM CS2 (Secondary Map)
I	EMIF ASRAM CS3 (Secondary Map)
J	EMIF ASRAM CS4 (Secondary Map)

The EMIF module is accessible by initiators other than the CPU. The CPU dataline buffer is updated on CPU reads only, and does not hold the latest data if another initiator updated the same address. For this reason, write accesses from other initiators are tracked locally to the dataline buffer. The dataline buffer is invalidated on the completion of a write access from another initiator if the tag matches the write address. In such a case, the arbitration scheme makes sure the write is performed to memory first to maintain data coherency. The corresponding dataline buffer of a CPU is also invalidated if the CPU enters a fault state. The CPU reads the old data until the time at which the buffer is invalidated.

14.3.3.1 CPU Write FIFO

The CPU write FIFO has entries for the access's Address[31:3], data, size/position of access (which bytes are valid), and whether the operation is part of an atomic access. This information is necessary to reconstruct the write access. The FIFO stalls the data write signal for subsequent CPU write accesses. The push logic routes incoming accesses to the arbiter if there is no outstanding write request. Otherwise, the entry is put onto the FIFO and outstanding accesses complete before popping the first-in entry onto the arbiter interface.

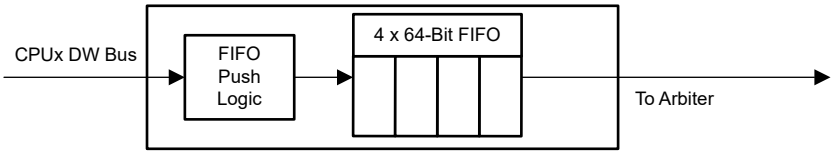


Figure 14-17. Write FIFO

14.3.4 Emulation Mode

Emulation mode is covered in the Debug Subsystem and the *Safety and Security Unit (SSU)* chapter.

14.4 Example Configuration

This section presents an example of interfacing the EMIF1 to both an SDR SDRAM device and an asynchronous Flash device.

14.4.1 Hardware Interface

Figure 14-18 shows the hardware interface between the EMIF, a Samsung K4S641632H-TC(L)70 64Mb SDRAM device, and a SHARP LH28F800BJE-PTTL90 8Mb Flash memory. The connection between EMIF and the SDRAM is straightforward, but the connection between the EMIF and the Flash deserves a detailed look.

The address inputs for the Flash are provided by three sources. The A[18:0] address inputs are provided by a combination of the EM1A and EM1BA pins according to Section 14.2.6.1, and a set of GPIO pins. The RD/nBY signal from Flash is connected to EM1WAIT pin of the EMIF.

Finally, this example configuration connects the $\overline{\text{EM1WE}}$ pin to the nWE input of the Flash and operates the EMIF in select strobe mode.

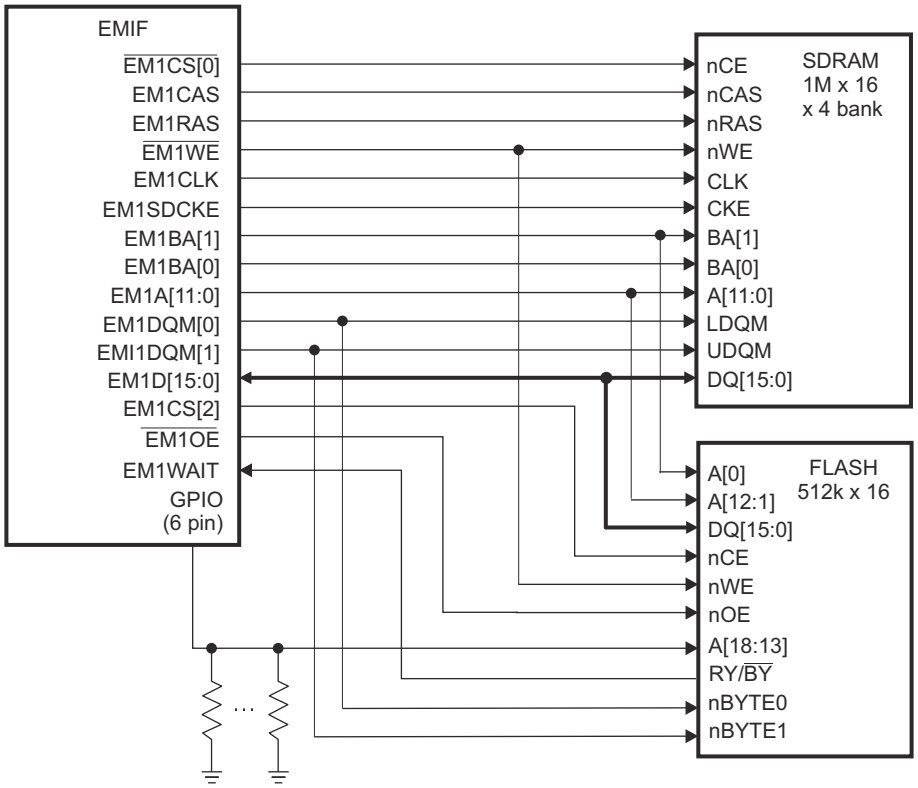


Figure 14-18. Example Configuration Interface

14.4.2 Software Configuration

The following sections describe how to configure the EMIF registers and bit fields to interface the EMIF with the Samsung K4S641632H-TC(L)70 SDRAM and the SHARP LH28F800BJE-PTTL90 8Mb Flash memory.

14.4.2.1 Configuring the SDRAM Interface

This section describes how to configure the EMIF to interface with the Samsung K4S641632H-TC(L)70 SDRAM with a clock frequency of f_{EM1CLK} = 100MHz. Procedure A described in Section 14.2.5.5 is followed that assumes that the SDRAM power-up timing constraints were met during the SDRAM auto-initialization sequence after reset.

14.4.2.1.1 PLL Programming for EMIF to K4S641632H-TC(L)70 Interface

If the system PLL is programmed to provide a SYSCLK frequency > 100MHz, then configure the EMIF1CLKDIV field in the PERSYSCLKDIVSEL register to make EM1CLK = SYSCLK/2 (default configuration). Before doing this, the SDRAM must be placed in self-refresh mode by setting the SR bit in the SDRAM configuration register. Once the EM1CLK frequency has been configured, remove the SDRAM from self-refresh by clearing the SR bit in SDRAM_CR.

Table 14-32. SR Field Value For EMIF to K4S641632H-TC(L)70 Interface

Field	Value	Purpose
SR	1 then 0	To place the EMIF into the self-refresh state

14.4.2.1.2 SDRAM Timing Register (SDRAM_TR) Settings for EMIF to K4S641632H-TC(L)70 Interface

The fields of the SDRAM timing register (SDRAM_TR) must be programmed first as described in Table 14-33 to satisfy the required timing parameters for the K4S641632H-TC(L)70. Based on these calculations, a value of 6111 4610h must be written to the SDRAM_TR. Figure 14-19 shows a graphical description of how the SDRAM_TR must be programmed.

Table 14-33. SDRAM_TR Field Calculations for EMIF to K4S641632H-TC(L)70 Interface

Field Name	Formula	Value from K4S641632H-TC(L)70 Data Sheet	Value Calculated for Field
T_RFC	T_RFC >= (t_{RFC} × f_{EM1CLK}) - 1	t_{RC} = 68ns (minimum)(1)	6
T_RP	T_RP >= (t_{RP} × f_{EM1CLK}) - 1	t_{RP} = 20ns (minimum)	1
T_RCD	T_RCD >= (t_{RCD} × f_{EM1CLK}) - 1	t_{RCD} = 20ns (minimum)	1
T_WR	T_WR >= (t_{WR} × f_{EM1CLK}) - 1	t_{RDL} = 2 CLK = 20ns (minimum)(2)	1
T_RAS	T_RAS >= (t_{RAS} × f_{EM1CLK}) - 1	t_{RAS} = 49ns (minimum)	4
T_RC	T_RC >= (t_{RC} × f_{EM1CLK}) - 1	t_{RC} = 68ns (minimum)	6
T_RRD	T_RRD >= (t_{RRD} × f_{EM1CLK}) - 1	t_{RRD} = 14ns (minimum)	1

(1) The Samsung data sheet does not specify a t_{RFC} value. Instead, Samsung specifies t_{RC} as the minimum auto refresh period.
(2) The Samsung data sheet does not specify a t_{WR} value. Instead, Samsung specifies t_{RDL} as last data in to row precharge minimum delay.

Figure 14-19. SDRAM Timing Register (SDRAM_TR)

31	30	29	28	27	26	24	23	22	21	20	19	18	17	16
0 0110					001		0	001			0	001		
T_RFC					T_RP		Rsvd	T_RCD			Rsvd	T_WR		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0100				0110				0	001			0000			
T_RAS				T_RC				Rsvd	T_RRD			Reserved			

14.4.2.1.3 SDRAM Self Refresh Exit Timing Register (SDR_EXT_TMNG) Settings for EMIF to K4S641632H-TC(L)70 Interface

The SDRAM self-refresh exit timing register (SDSRETR) must be programmed second to satisfy the t_{XSR} timing requirement from the K4S641632H-TC(L)70 data sheet. Table 14-34 shows the calculation of the proper value to program into the T_XS field of this register. Based on this calculation, a value of 6h must be written to the SDSRETR. Figure 14-20 shows how the SDSRETR must be programmed.

Table 14-34. RR Calculation for EMIF to K4S641632H-TC(L)70 Interface

Field Name	Formula	Value from K4S641632H-TC(L)70 Data Sheet	Value Calculated for Field
T_XS	T_XS >= (t_{XSR} × f_{EM1CLK}) - 1	t_{RC} = 68ns (minimum)(1)	6

(1) The Samsung data sheet does not specify a t_{XSR} value. Instead, Samsung specifies t_{RC} as the minimum required time after CKE going high to complete self-refresh exit.

Figure 14-20. SDRAM Self Refresh Exit Timing Register (SDR_EXT_TMNG)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0000 0000 0000 0000															
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000 0000 0000											0 0110				
Reserved											T_XS				

14.4.2.1.4 SDRAM Refresh Control Register (SDRAM_RCR) Settings for EMIF to K4S641632H-TC(L)70 Interface

The SDRAM refresh control register (SDRAM_RCR) can next be programmed to satisfy the required refresh rate of the K4S641632H-TC(L)70. Table 14-35 shows the calculation of the proper value to program into the RR field of this register. Based on this calculation, a value of 61Ah must be written to the SDRAM_RCR. Figure 14-21 shows how the SDRAM_RCR must be programmed.

Table 14-35. RR Calculation for EMIF to K4S641632H-TC(L)70 Interface

Field Name	Formula	Values	Value Calculated for Field
RR	RR ≤ f_{EM1CLK} × $t_{Refresh\ Period}$ / n_{cycles}	From SDRAM data sheet: $t_{Refresh\ Period}$ = 64ms; n_{cycles} = 4096 EMIF clock rate: f_{EM1CLK} = 100MHz	RR = 1562 cycles = 61Ah cycles

Figure 14-21. SDRAM Refresh Control Register (SDRAM_RCR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0 0000 0000 0000														000	
Reserved														Reserved	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000			0 0110 0001 1010 (61Ah)												
Reserved			RR												

14.4.2.1.5 SDRAM Configuration Register (SDRAM_CR) Settings for EMIF to K4S641632H-TC(L)70 Interface

Finally, the fields of the SDRAM configuration register (SDRAM_CR) must be programmed as described in Table 14-36 to properly interface with the K4S641632H-TC(L)70 device. Based on these settings, a value of 4720h must be written to the SDRAM_CR. Figure 14-22 shows how the SDRAM_CR must be programmed. The EMIF is now ready to perform read and write accesses to the SDRAM.

Table 14-36. SDRAM_CR Field Values For EMIF to K4S641632H-TC(L)70 Interface

Field	Value	Purpose
SR	0	To avoid placing the EMIF into the self-refresh state
NM	1	To configure the EMIF for a 16-bit data bus
CL	011b	To select a CAS latency of 3
BIT11_9LOCK	1	To allow the CL field to be written
IBANK	010b	To select 4 internal SDRAM banks
PAGESIZE	0	To select a page size of 256 words

Figure 14-22. SDRAM Configuration Register (SDRAM_CR)

31	30	29	28	27	26	25	24
0	0	0	0 0000				
SR	Reserved	Reserved	Reserved				

23	22	21	20	19	18	17	16
00 0000						0	0
Reserved						Reserved	Reserved

15	14	13	12	11	10	9	8
0	1	0	0	011			1
Reserved	NM	Reserved	Reserved	CL			BIT11_9LOCK

7	6	5	4	3	2	1	0
0	010			0	000		
Reserved	IBANK			Reserved	PAGESIZE		

14.4.2.2 Configuring the Flash Interface

This section describes how to configure the EMIF to interface with the SHARP LH28F800BJE-PTTL90 8Mb Flash memory with a clock frequency of f_{EM1CLK} = 100MHz.

14.4.2.2.1 Asynchronous 1 Configuration Register (ASYNC_CS2_CFG) Settings for EMIF to LH28F800BJE-PTTL90 Interface

The asynchronous 1 configuration register (ASYNC_CS2_CFG) is the only register that is necessary to program for this asynchronous interface. The SS bit must be set to 1 to enable select strobe command and the ASIZE field can be set to 1 to select a 16-bit interface. The other fields in this register control the shaping of the EMIF signals, and the proper values can be determined by referring to the AC Characteristics in the Flash data sheet and the device data sheet. Table 14-37 and Table 14-38 show the pertinent AC Characteristics for reads and writes to the Flash device, and Figure 14-23 and Figure 14-24 show the associated timing waveforms. Finally, Figure 14-25 shows programming the ASYNC_CS2_CFG with the calculated values.

Table 14-37. AC Characteristics for a Read Access

AC Characteristic	Device	Definition	Minimum	Maximum	Unit
t_{SU}	EMIF	Setup time, read EM1D before EM1OE high	15		ns
t_{H}	EMIF	Data hold time, read EM1D after EM1OE high	0		ns
t_{ELQV}	Flash	nCE to Output Delay		90	ns
t_{EHQZ}	Flash	nCE High to Output in High Impedance		55	ns

Table 14-38. AC Characteristics for a Write Access

AC Characteristic	Device	Definition	Minimum	Maximum	Unit
t_{AVAV}	Flash	Write Cycle Time	90		ns
t_{ELEH}	Flash	nCE Pulse Width Low	50		ns
t_{EHEL}	Flash	nCE Pulse Width High (not shown in Figure 14-24)	30		ns

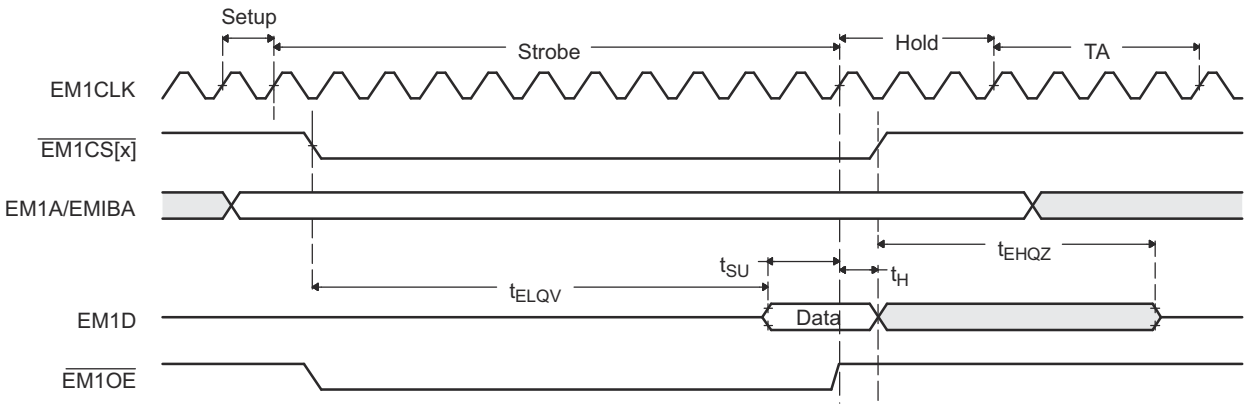


Figure 14-23. LH28F800BJE-PTTL90 to EMIF Read Timing Waveforms

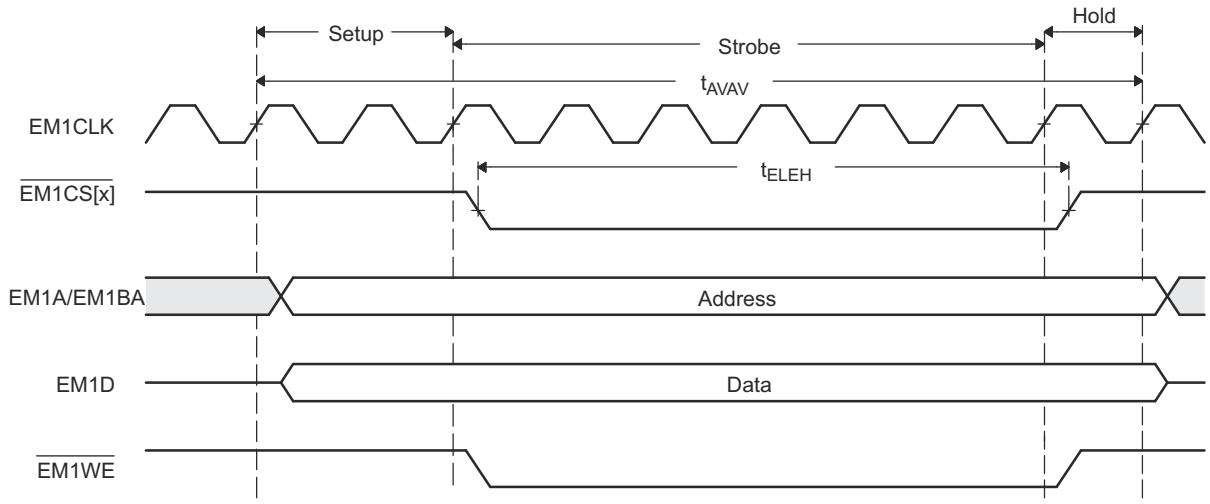


Figure 14-24. LH28F800BJE-PTTL90 to EMIF Write Timing Waveforms

The R_STROBE field must be set to meet the following equation:

$R\_STROBE >= (t_{ELQV} + t_{SU}) \times f_{EM1CLK} - 1$

R_STROBE >= (90ns + 15ns) × 200MHz - 1

R_STROBE >= 20

R_STROBE = 20

The R_HOLD field must be large enough to satisfy EMIF Data hold time, t_H:

$R\_HOLD > = t_H \times f_{EM1CLK} - 1$

R_HOLD >= 0ns × 200MHz - 1

R_HOLD >= -1

The R_HOLD field must also combine with the TA field to satisfy the Flash nCE High to Output in High Impedance time, t_{EHQZ}:

$R\_HOLD + TA >= t_{EHQZ} \times f_{EM1CLK} - 2$

R_HOLD + TA >= 55ns × 200MHz - 2

R_HOLD + TA >= 9

The largest value that can be programmed into the TA field is 3h, therefore the following values must be used:

R_HOLD = 6

TA = 3

For Writes, the W_STROBE field must be set to satisfy the Flash nCE Pulse Width constraint, t_{ELEH}:

$W\_STROBE >= t_{ELEH} \times f_{EM1CLK} - 1$

W_STROBE >= 50ns × 200MHz - 1

W_STROBE >= 9

The W_SETUP and W_HOLD fields must combine to satisfy the Flash nCE Pulse Width High constraint, t_{EHEL}:

W_SETUP + W_HOLD > = t_{EHEL} × f_{EM1CLK} - 2

W_SETUP + W_HOLD > = 30ns × 200MHz - 2

W_SETUP + W_HOLD > = 4

In addition, the entire Write access length must satisfy the Flash minimum Write Cycle Time, t_{AVAV}:

W_SETUP + W_STROBE + W_HOLD >= t_{AVAV} × f_{EM1CLK} - 3

W_SETUP + W_STROBE + W_HOLD >= 90ns × 200MHz - 3

W_SETUP + W_STROBE + W_HOLD >= 15

Solving the above equations for the Write fields results in the following:

W_SETUP = 4

W_STROBE = 10

W_HOLD = 1

Adding a 5ns (1 cycle) margin to each of the periods (excluding TA that is already at the maximum) in this example produces the following recommended values:

W_SETUP = 5h

W_STROBE = Bh

W_HOLD = 2h

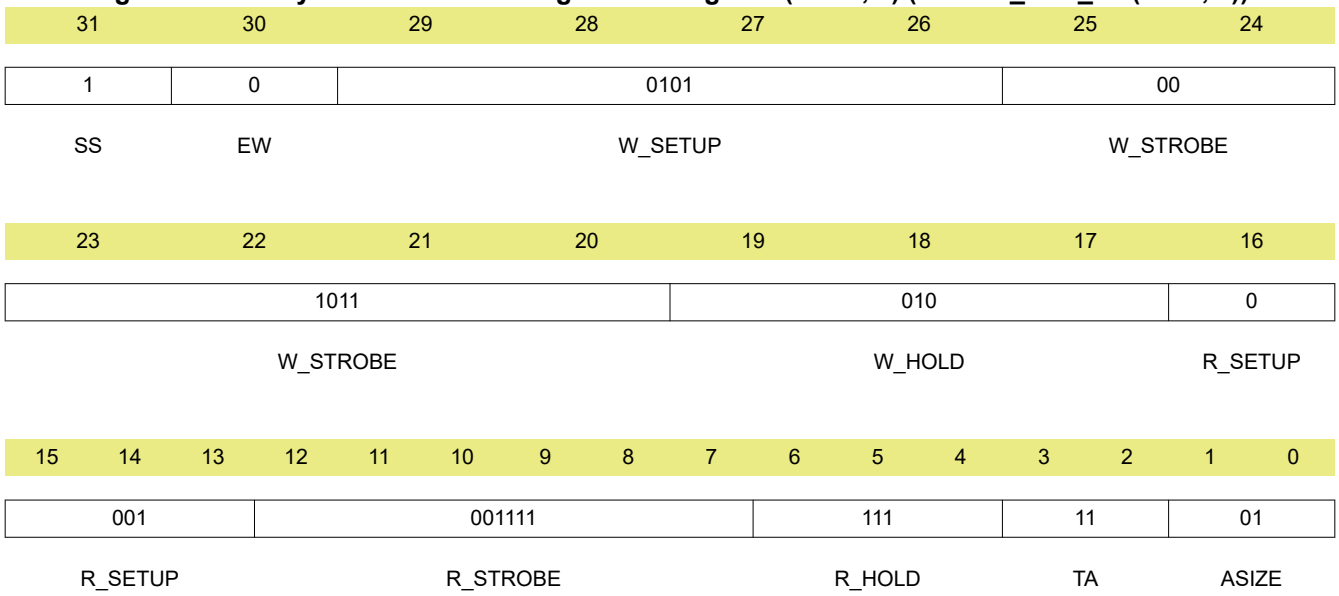
R_SETUP = 1h

R_STROBE = 15h

R_HOLD = 7h

TA = 3h

Figure 14-25. Asynchronous *m* Configuration Register (*m* = 1, 2) (ASYNC_CS*n*_CR(*n* = 2, 3))



14.5 Software

14.5.1 EMIF Registers to Driverlib Functions

Table 14-39. EMIF Registers to Driverlib Functions

File	Driverlib Function
RCSR	
-	
ASYNC_WCCR	
-	
SDRAM_CR	
-	
SDRAM_RCR	
-	
ASYNC_CS2_CR	
-	
ASYNC_CS3_CR	
-	See ASYNC_CS2_CR
ASYNC_CS4_CR	
-	See ASYNC_CS2_CR
SDRAM_TR	
-	
TOTAL_SDRAM_AR	
-	
TOTAL_SDRAM_ACTR	
-	
SDR_EXT_TMNG	
-	
INT_RAW	
-	
INT_MSK	
-	
INT_MSK_SET	
-	
INT_MSK_CLR	
-	

14.5.2 EMIF Examples

NOTE: These examples are located in the C2000Ware installation at the following location: C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/emif

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

14.6 EMIF Registers

This section describes the EMIF Registers.

14.6.1 EMIF Base Address Table

Table 14-40. EMIF Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
EMIF_REGS	EMIF1_BASE	0x3080_0000	-	YES	YES	YES	-	-	-	YES

14.6.2 EMIF_REGS Registers

Table 14-41 lists the memory-mapped registers for the EMIF_REGS registers. All register offset addresses not listed in Table 14-41 should be considered as reserved locations and the register contents should not be modified.

Table 14-41. EMIF_REGS Registers

Offset	Acronym	Register Name	Protection
0h	RCSR	Revision Code and Status Register	
4h	ASYNC_WCCR	Async Wait Cycle Config Register	
8h	SDRAM_CR	SDRAM (EMxCS0n) Config Register	
Ch	SDRAM_RCR	SDRAM Refresh Control Register	
10h	ASYNC_CS2_CR	Async 1 (EMxCS2n) Config Register	
14h	ASYNC_CS3_CR	Async 2 (EMxCS3n) Config Register	
18h	ASYNC_CS4_CR	Async 3 (EMxCS4n) Config Register	
20h	SDRAM_TR	SDRAM Timing Register	
30h	TOTAL_SDRAM_AR	Total SDRAM Accesses Register	
34h	TOTAL_SDRAM_ACTR	Total SDRAM Activate Register	
3Ch	SDR_EXT_TMNG	SDRAM SR/PD Exit Timing Register	
40h	INT_RAW	Interrupt Raw Register	
44h	INT_MSK	Interrupt Masked Register	
48h	INT_MSK_SET	Interrupt Mask Set Register	
4Ch	INT_MSK_CLR	Interrupt Mask Clear Register	

Complex bit access types are encoded to fit into small table cells. Table 14-42 shows the codes that are used for access types in this section.

Table 14-42. EMIF_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

14.6.2.1 RCSR Register (Offset = 0h) [Reset = 40000205h]

RCSR is shown in Figure 14-26 and described in Table 14-43.

Return to the Summary Table.

Revision Code and Status Register

Figure 14-26. RCSR Register

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

Table 14-43. RCSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	BE	R	0h	EMIF endian mode. 0: Little Endian. 1: Big Endian. Reset type: SYSRSn
30	FR	R	1h	EMIF operating rate. 0: Half Rate. 1: Full Rate. Reset type: SYSRSn
29-16	MODULE_ID	R	0h	EMIF module ID. 0x0000: EMIF_24. 0x000E: EMIF_24 SDRAM. 0x000F: EMIF_24 ASYNC. Reset type: SYSRSn
15-8	MAJOR_REVISION	R	2h	Major Revision. EMIF code revisions are indicated by a revision code taking the format major_revision.minor_revision. Reset type: SYSRSn
7-0	MINOR_REVISION	R	5h	Minor Revision. See major_revision field description. Reset type: SYSRSn

14.6.2.2 ASYNC_WCCR Register (Offset = 4h) [Reset = F0000080h]

ASYNC_WCCR is shown in Figure 14-27 and described in Table 14-44.

Return to the Summary Table.

Async Wait Cycle Config Register

Figure 14-27. ASYNC_WCCR Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	WP0	RESERVED			
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R-0h			
23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
MAX_EXT_WAIT							
R/W-80h							

Table 14-44. ASYNC_WCCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	1h	Reserved
30	RESERVED	R/W	1h	Reserved
29	RESERVED	R/W	1h	Reserved
28	WP0	R/W	1h	Defines the polarity of the EMxWAIT port.: 0: Wait if EMxWAIT port is low. 1: Wait if EMxWAIT port is high. Reset type: SYSRSn
27-24	RESERVED	R	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-8	RESERVED	R	0h	Reserved
7-0	MAX_EXT_WAIT	R/W	80h	The EMIF will wait for (max_ext_wait + 1) * 16 clock cycles before an extended asynchronous cycle is terminated. Reset type: SYSRSn

14.6.2.3 SDRAM_CR Register (Offset = 8h) [Reset = 00000620h]

SDRAM_CR is shown in Figure 14-28 and described in Table 14-45.

Return to the Summary Table.

SDRAM (EMxCS0n) Config Register

Figure 14-28. SDRAM_CR Register

31	30	29	28	27	26	25	24
SR	PD	PDWR	RESERVED			RESERVED	
R/W-0h	R/W-0h	R/W-0h	R-0h			R/W-0h	
23	22	21	20	19	18	17	16
RESERVED	RESERVED			RESERVED	RESERVED		RESERVED
R/W-0h	R/W-0h			R/W-0h	R/W-0h		R/W-0h
15	14	13	12	11	10	9	8
RESERVED	NM	RESERVED	RESERVED	CL			BIT_11_9_LOCK
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-3h			R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	IBANK			RESERVED	PAGESIGE		
R-0h	R/W-2h			R/W-0h	R/W-0h		

Table 14-45. SDRAM_CR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SR	R/W	0h	Self Refresh. Writing a 1 to this bit will cause connected SDRAM devices to be placed into Self Refresh mode and the EMIF to enter the self refresh state. In this state the EMIF will service all asynchronous memory accesses immediately but any SDRAM access will take at least t_ras + 1 cycles due to the time required for the SDRAM devices to out of Self Refresh mode. If an SDRAM access immediately follows the setting of the sr bit, the access will take t_ras + t_xs + 2 cycles. If both sr and pd bits are set, the EMIF will go into Self Refresh. Reset type: SYSRSn
30	PD	R/W	0h	Power Down. Writing a 1 to this bit will cause connected SDRAM devices to be placed into Power Down mode. If both sr and pd bits are set, the EMIF will go into Self Refresh. Reset type: SYSRSn
29	PDWR	R/W	0h	Perform refreshes during Power Down. Writing a 1 to this bit will cause the EMIF to exit the power down state and issue an AUTO REFRESH command every time Refresh May level is set. Reset type: SYSRSn
28-26	RESERVED	R	0h	Reserved
25-23	RESERVED	R/W	0h	Reserved
22-20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18-17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R	0h	Reserved

Table 14-45. SDRAM_CR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	NM	R/W	0h	Narrow mode. Set to 1 when system bus width to memory bus width is 2:1 for SDR SDRAM. Set to 0 when system bus width to memory bus width is 1:1 for SDR SDRAM. A write to this field will cause the EMIF to start the SDRAM initialization sequence. Reset type: SYSRSn
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11-9	CL	R/W	3h	The value of this field defines the CAS latency to be used when accessing connected SDRAM devices. Only CAS latencies of 2 (cl = 2) and 3 (cl = 3) are supported. A write to this field will cause the EMIF to start the SDRAM initialization sequence. Reset type: SYSRSn
8	BIT_11_9_LOCK	R-0/W1S	0h	Bits 11 to 9 can only be written if this bit is set to 1. Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6-4	IBANK	R/W	2h	Defines number of banks inside connected SDRAM devices: 000: 1 bank SDRAM devices. 001: 2 bank SDRAM devices. 010: 4 bank SDRAM devices. 011: Reserved. 1xx: Reserved. A write to this field will cause the EMIF to start the SDRAM initialization sequence. Reset type: SYSRSn
3	RESERVED	R/W	0h	Reserved
2-0	PAGESIGE	R/W	0h	Defines the internal page size of connected SDRAM devices: 000: 256-word pages requiring 8 column address bits. 001: 512-word pages requiring 9 column address bits. 010: 1024-word pages requiring 10 column address bits. 011: 2048-word pages requiring 11 column address bits. 1xx: Reserved. A write to this field will cause the EMIF to start the SDRAM initialization sequence. Reset type: SYSRSn

14.6.2.4 SDRAM_RCR Register (Offset = Ch) [Reset = 00000080h]

SDRAM_RCR is shown in Figure 14-29 and described in Table 14-46.

Return to the Summary Table.

SDRAM Refresh Control Register

Figure 14-29. SDRAM_RCR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED					RESERVED		
R-0h					R/W-0h		
15	14	13	12	11	10	9	8
RESERVED			REFRESH_RATE				
R-0h			R/W-80h				
7	6	5	4	3	2	1	0
REFRESH_RATE							
R/W-80h							

Table 14-46. SDRAM_RCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18-16	RESERVED	R/W	0h	Reserved
15-13	RESERVED	R	0h	Reserved
12-0	REFRESH_RATE	R/W	80h	Value in this field is used to define the rate at which connected SDRAM devices will be refreshed, as follows: SDRAM refresh rate = EMIF rate/refresh_rate where EMIF rate=clk rate when full_rate=1, or EMIF rate=1/2 clk rate when full_rate=0. Writing a value < 0x0020 to this field will cause it to be loaded with (2 * t_rfc) + 1 value from SDRAM Timing register. Reset type: SYSRSn

14.6.2.5 ASYNC_CS2_CR Register (Offset = 10h) [Reset = 3FFFFFFDh]

ASYNC_CS2_CR is shown in Figure 14-30 and described in Table 14-47.

Return to the Summary Table.

Async 1 (EMxCS2n) Config Register

Figure 14-30. ASYNC_CS2_CR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SS	EW	W_SETUP				W_STROBE						W_HOLD			R_SETUP
R/W-0h	R/W-0h	R/W-Fh				R/W-3Fh						R/W-7h			R/W-Fh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R_SETUP			R_STROBE						R_HOLD			TA		ASIZE	
R/W-Fh			R/W-3Fh						R/W-7h			R/W-3h		R/W-1h	

Table 14-47. ASYNC_CS2_CR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SS	R/W	0h	Select Strobe mode. Set to 1 if chip selects need to have write or read strobe timing. Reset type: SYSRSn
30	EW	R/W	0h	Extend Wait mode. Set to 1 if extended asynchronous cycles are required based on EMxWAIT. Reset type: SYSRSn
29-26	W_SETUP	R/W	Fh	Write Strobe Setup cycles. Number of EMxCLK cycles from EMxAy, EMxBAy, EMxDQMy, and EMxCS2n being set to EMxWEn asserted, minus one cycle. The reset value is 16 cycles. Reset type: SYSRSn
25-20	W_STROBE	R/W	3Fh	Write Strobe Duration cycles. Number of EMxCLK cycles for which EMxWEn is held active, minus one cycle. The reset value is 64 cycles. This field cannot be zero when ew = 1. Reset type: SYSRSn
19-17	W_HOLD	R/W	7h	Write Strobe Hold cycles. Number of EMxCLK cycles for which EMxAy, EMxBAy, EMxDQMy, and EMxCS2n are held after EMxWEn has been deasserted, minus one cycle. The reset value is 8 cycles. Reset type: SYSRSn
16-13	R_SETUP	R/W	Fh	Read Strobe Setup cycles. Number of EMxCLK cycles from EMxAy, EMxBAy, EMxDQMy, and EMxCS2n being set to EMxOEn asserted, minus one cycle. The reset value is 16 cycles. Reset type: SYSRSn
12-7	R_STROBE	R/W	3Fh	Read Strobe Duration cycles. Number of EMxCLK cycles for which EMxOEn is held active, minus one cycle. The reset value is 64 cycles. This field cannot be zero when ew = 1. Reset type: SYSRSn
6-4	R_HOLD	R/W	7h	Read Strobe Hold cycles. Number of EMxCLK cycles for which EMxAy, EMxBAy, EMxDQMy, and EMxCS2n are held after EMxOEn has been deasserted, minus one cycle. The reset value is 8 cycles. Reset type: SYSRSn
3-2	TA	R/W	3h	Turn Around cycles. Number of EMxCLK cycles between the end of one asynchronous memory access and the start of another asynchronous memory access, minus one cycle. This delay is not incurred between a read followed by a read, or a write followed by a write to the same chip select. The reset value is 4 cycles. Reset type: SYSRSn

Table 14-47. ASYNC_CS2_CR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	ASIZE	R/W	1h	Asynchronous Memory Size. Defines the width of the asynchronous device's data bus : 00: 8 Bit data bus. 01: 16 Bit data bus. 10: 32 Bit data bus. 11: Reserved. Reset type: SYSRSn

14.6.2.6 ASYNC_CS3_CR Register (Offset = 14h) [Reset = 3FFFFFFDh]

ASYNC_CS3_CR is shown in Figure 14-31 and described in Table 14-48.

Return to the Summary Table.

Async 2 (EMxCS3n) Config Register

Figure 14-31. ASYNC_CS3_CR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SS	EW	W_SETUP				W_STROBE						W_HOLD			R_SETUP
R/W-0h	R/W-0h	R/W-Fh				R/W-3Fh						R/W-7h			R/W-Fh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R_SETUP			R_STROBE						R_HOLD			TA		ASIZE	
R/W-Fh			R/W-3Fh						R/W-7h			R/W-3h		R/W-1h	

Table 14-48. ASYNC_CS3_CR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SS	R/W	0h	Select Strobe mode. Set to 1 if chip selects need to have write or read strobe timing. Reset type: SYSRSn
30	EW	R/W	0h	Extend Wait mode. Set to 1 if extended asynchronous cycles are required based on EMxWAIT. Reset type: SYSRSn
29-26	W_SETUP	R/W	Fh	Write Strobe Setup cycles. Number of EMxCLK cycles from EMxAy, EMxBAy, EMxDQMy, and EMxCS3n being set to EMxWEn asserted, minus one cycle. The reset value is 16 cycles. Reset type: SYSRSn
25-20	W_STROBE	R/W	3Fh	Write Strobe Duration cycles. Number of EMxCLK cycles for which EMxWEn is held active, minus one cycle. The reset value is 64 cycles. This field cannot be zero when ew = 1. Reset type: SYSRSn
19-17	W_HOLD	R/W	7h	Write Strobe Hold cycles. Number of EMxCLK cycles for which EMxAy, EMxBAy, EMxDQMy, and EMxCS3n are held after EMxWEn has been deasserted, minus one cycle. The reset value is 8 cycles. Reset type: SYSRSn
16-13	R_SETUP	R/W	Fh	Read Strobe Setup cycles. Number of EMxCLK cycles from EMxAy, EMxBAy, EMxDQMy, and EMxCS3n being set to EMxOEn asserted, minus one cycle. The reset value is 16 cycles. Reset type: SYSRSn
12-7	R_STROBE	R/W	3Fh	Read Strobe Duration cycles. Number of EMxCLK cycles for which EMxOEn is held active, minus one cycle. The reset value is 64 cycles. This field cannot be zero when ew = 1. Reset type: SYSRSn
6-4	R_HOLD	R/W	7h	Read Strobe Hold cycles. Number of EMxCLK cycles for which EMxAy, EMxBAy, EMxDQMy, and EMxCS3n are held after EMxOEn has been deasserted, minus one cycle. The reset value is 8 cycles. Reset type: SYSRSn
3-2	TA	R/W	3h	Turn Around cycles. Number of EMxCLK cycles between the end of one asynchronous memory access and the start of another asynchronous memory access, minus one cycle. This delay is not incurred between a read followed by a read, or a write followed by a write to the same chip select. The reset value is 4 cycles. Reset type: SYSRSn

Table 14-48. ASYNC_CS3_CR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	ASIZE	R/W	1h	Asynchronous Memory Size. Defines the width of the asynchronous device's data bus : 00: 8 Bit data bus. 01: 16 Bit data bus. 10: 32 Bit data bus. 11: Reserved. Reset type: SYSRSn

14.6.2.7 ASYNC_CS4_CR Register (Offset = 18h) [Reset = 3FFFFFFDh]

ASYNC_CS4_CR is shown in Figure 14-32 and described in Table 14-49.

Return to the Summary Table.

Async 3 (EMxCS4n) Config Register

Figure 14-32. ASYNC_CS4_CR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SS	EW	W_SETUP				W_STROBE						W_HOLD			R_SETUP
R/W-0h	R/W-0h	R/W-Fh				R/W-3Fh						R/W-7h			R/W-Fh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R_SETUP			R_STROBE						R_HOLD			TA		ASIZE	
R/W-Fh			R/W-3Fh						R/W-7h			R/W-3h		R/W-1h	

Table 14-49. ASYNC_CS4_CR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SS	R/W	0h	Select Strobe mode. Set to 1 if chip selects need to have write or read strobe timing. Reset type: SYSRSn
30	EW	R/W	0h	Extend Wait mode. Set to 1 if extended asynchronous cycles are required based on EMxWAIT. Reset type: SYSRSn
29-26	W_SETUP	R/W	Fh	Write Strobe Setup cycles. Number of EMxCLK cycles from EMxAy, EMxBAy, EMxDQMy, and EMxCS4n being set to EMxWEn asserted, minus one cycle. The reset value is 16 cycles. Reset type: SYSRSn
25-20	W_STROBE	R/W	3Fh	Write Strobe Duration cycles. Number of EMxCLK cycles for which EMxWEn is held active, minus one cycle. The reset value is 64 cycles. This field cannot be zero when ew = 1. Reset type: SYSRSn
19-17	W_HOLD	R/W	7h	Write Strobe Hold cycles. Number of EMxCLK cycles for which EMxAy, EMxBAy, EMxDQMy, and EMxCS4n are held after EMxWEn has been deasserted, minus one cycle. The reset value is 8 cycles. Reset type: SYSRSn
16-13	R_SETUP	R/W	Fh	Read Strobe Setup cycles. Number of EMxCLK cycles from EMxAy, EMxBAy, EMxDQMy, and EMxCS4n being set to EMxOEn asserted, minus one cycle. The reset value is 16 cycles. Reset type: SYSRSn
12-7	R_STROBE	R/W	3Fh	Read Strobe Duration cycles. Number of EMxCLK cycles for which EMxOEn is held active, minus one cycle. The reset value is 64 cycles. This field cannot be zero when ew = 1. Reset type: SYSRSn
6-4	R_HOLD	R/W	7h	Read Strobe Hold cycles. Number of EMxCLK cycles for which EMxAy, EMxBAy, EMxDQMy, and EMxCS4n are held after EMxOEn has been deasserted, minus one cycle. The reset value is 8 cycles. Reset type: SYSRSn
3-2	TA	R/W	3h	Turn Around cycles. Number of EMxCLK cycles between the end of one asynchronous memory access and the start of another asynchronous memory access, minus one cycle. This delay is not incurred between a read followed by a read, or a write followed by a write to the same chip select. The reset value is 4 cycles. Reset type: SYSRSn

Table 14-49. ASYNC_CS4_CR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	ASIZE	R/W	1h	Asynchronous Memory Size. Defines the width of the asynchronous device's data bus : 00: 8 Bit data bus. 01: 16 Bit data bus. 10: 32 Bit data bus. 11: Reserved. Reset type: SYSRSn

14.6.2.8 SDRAM_TR Register (Offset = 20h) [Reset = 19214610h]

SDRAM_TR is shown in Figure 14-33 and described in Table 14-50.

Return to the Summary Table.

SDRAM Timing Register

Figure 14-33. SDRAM_TR Register

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

Table 14-50. SDRAM_TR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	T_RFC	R/W	3h	Minimum number of EMxCLK cycles from Refresh or Load Mode to Refresh or Activate, minus one. Reset type: SYSRSn
26-24	T_RP	R/W	1h	Minimum number of EMxCLK cycles from Precharge to Activate or Refresh, minus one. Reset type: SYSRSn
23	RESERVED	R	0h	Reserved
22-20	T_RCD	R/W	2h	Minimum number of EMxCLK cycles from Activate to Read or Write, minus one. Reset type: SYSRSn
19	RESERVED	R	0h	Reserved
18-16	T_WR	R/W	1h	For SDR, this is equal to minimum number of EMxCLK cycles from last Write transfer to Precharge, minus one. Reset type: SYSRSn
15-12	T_RAS	R/W	4h	Minimum number of EMxCLK cycles from Activate to Precharge, minus one. t_ras >= t_rcd. Reset type: SYSRSn
11-8	T_RC	R/W	6h	Minimum number of EMxCLK cycles from Activate to Activate minus one. Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6-4	T_RRD	R/W	1h	Minimum number of EMxCLK cycles from Activate to Activate for a different bank, minus one. Reset type: SYSRSn
3-0	RESERVED	R	0h	Reserved

14.6.2.9 TOTAL_SDRAM_AR Register (Offset = 30h) [Reset = 00000000h]

TOTAL_SDRAM_AR is shown in Figure 14-34 and described in Table 14-51.

Return to the Summary Table.

Total SDRAM Accesses Register

Figure 14-34. TOTAL_SDRAM_AR Register

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

Table 14-51. TOTAL_SDRAM_AR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TOTAL_SDRAM_AR	R	0h	Indicates the total number of accesses to SDRAM from a controller (CPUx/CPUX.DMA). This counter is incremented by two for a single access crossing page boundaries. Reset type: SYSRSn

14.6.2.10 TOTAL_SDRAM_ACTR Register (Offset = 34h) [Reset = 00000000h]

TOTAL_SDRAM_ACTR is shown in Figure 14-35 and described in Table 14-52.

Return to the Summary Table.

Total SDRAM Activate Register

Figure 14-35. TOTAL_SDRAM_ACTR Register

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

Table 14-52. TOTAL_SDRAM_ACTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TOTAL_SDRAM_ACTR	R	0h	Indicates the total number of SDRAM accesses which require an activate command. Reset type: SYSRSn

14.6.2.11 SDR_EXT_TMNG Register (Offset = 3Ch) [Reset = 00000007h]

SDR_EXT_TMNG is shown in Figure 14-36 and described in Table 14-53.

Return to the Summary Table.

SDRAM SR/PD Exit Timing Register

Figure 14-36. SDR_EXT_TMNG Register

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

Table 14-53. SDR_EXT_TMNG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-5	RESERVED	R	0h	Reserved
4-0	T_XS	R/W	7h	This is equal to minimum number of EMxCLK cycles from Self Refresh exit to any command, minus one. For SDR SDRAM, this count should satisfy tXSR. Reset type: SYSRSn

14.6.2.12 INT_RAW Register (Offset = 40h) [Reset = 00000000h]

INT_RAW is shown in Figure 14-37 and described in Table 14-54.

Return to the Summary Table.

Interrupt Raw Register

Figure 14-37. INT_RAW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										WR				LT	AT
R-0h										R/W1S-0h				R/W1S-0h	R/W1S-0h

Table 14-54. INT_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-6	RESERVED	R	0h	Reserved
5-2	WR	R/W1S	0h	Wait Rise. Set to 1 by hardware to indicate rising edge on the corresponding EMxWAIT has been detected. The WPx bits in the Async Wait Cycle Config register has no effect on these bits. Writing a 1 will clear these bits as well as the wr_masked bits in the Interrupt Masked register. Writing a 0 has no effect. Reset type: SYSRSn
1	LT	R/W1S	0h	Line Trap. Set to 1 by hardware to indicate illegal memory access type or invalid cache line size. Writing a 1 will clear this bit as well as the lt_masked bit in the Interrupt Masked register. Writing a 0 has no effect. Reset type: SYSRSn
0	AT	R/W1S	0h	Asynchronous Timeout. Set to 1 by hardware to indicate that during an extended asynchronous memory access cycle, the EMxWAIT signal did not go inactive within the number of cycles defined by the max_ext_wait field in Async Wait Cycle Config register. Writing a 1 will clear this bit as well as the at_masked bit in the Interrupt Masked register. Writing a 0 has no effect. Reset type: SYSRSn

14.6.2.13 INT_MSK Register (Offset = 44h) [Reset = 00000000h]

INT_MSK is shown in Figure 14-38 and described in Table 14-55.

Return to the Summary Table.

Interrupt Masked Register

Figure 14-38. INT_MSK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		WR_MASKED				LT_MASKED	AT_MASKED
R-0h		R/W1S-0h				R/W1S-0h	R/W1S-0h

Table 14-55. INT_MSK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-6	RESERVED	R	0h	Reserved
5-2	WR_MASKED	R/W1S	0h	Masked Wait Rise. Set to 1 by hardware to indicate rising edge on the corresponding EMxWAIT has been detected, only if the wr_mask_set bit in the Interrupt Mask Set register is set to 1. The WPx bits in the Async Wait Cycle Config register has no effect on these bits. Writing a 1 will clear these bits as well as the wr bits in the Interrupt Raw register. Writing a 0 has no effect. Reset type: SYSRSn
1	LT_MASKED	R/W1S	0h	Masked Line Trap. Set to 1 by hardware to indicate illegal memory access type or invalid cache line size, only if the lt_mask_set bit in the Interrupt Mask Set register is set to 1. Writing a 1 will clear this bit as well as the lt bit in the Interrupt Raw register. Writing a 0 has no effect. Reset type: SYSRSn
0	AT_MASKED	R/W1S	0h	Masked Asynchronous Timeout. Set to 1 by hardware to indicate that during an extended asynchronous memory access cycle, the EMxWAIT signal did not go inactive within the number of cycles defined by the max_ext_wait field in Async Wait Cycle Config register, only if the at_mask_set bit in the Interrupt Mask Set register is set to 1. Writing a 1 will clear this bit as well as the at bit in the Interrupt Raw register. Writing a 0 has no effect. Reset type: SYSRSn

14.6.2.14 INT_MSK_SET Register (Offset = 48h) [Reset = 00000000h]

INT_MSK_SET is shown in Figure 14-39 and described in Table 14-56.

Return to the Summary Table.

Interrupt Mask Set Register

Figure 14-39. INT_MSK_SET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		WR_MASK_SET				LT_MASK_SET	AT_MASK_SET
R-0h		R/W1S-0h				R/W1S-0h	R/W1S-0h

Table 14-56. INT_MSK_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-6	RESERVED	R	0h	Reserved
5-2	WR_MASK_SET	R/W1S	0h	Mask set for wr_masked bits in the Interrupt Masked Register. Writing a 1 will enable the interrupts, and set these bits as well as the wr_mask_clr bits in the Interrupt Mask Clear register. Writing a 0 has no effect. Reset type: SYSRSn
1	LT_MASK_SET	R/W1S	0h	Mask set for lt_masked bit in the Interrupt Masked Register. Writing a 1 will enable the interrupt, and set this bit as well as the lt_mask_clr bit in the Interrupt Mask Clear register. Writing a 0 has no effect. Reset type: SYSRSn
0	AT_MASK_SET	R/W1S	0h	Mask set for at_masked bit in the Interrupt Masked Register. Writing a 1 will enable the interrupt, and set this bit as well as the at_mask_clr bit in the Interrupt Mask Clear register. Writing a 0 has no effect. Reset type: SYSRSn

14.6.2.15 INT_MSK_CLR Register (Offset = 4Ch) [Reset = 00000000h]

INT_MSK_CLR is shown in Figure 14-40 and described in Table 14-57.

Return to the Summary Table.

Interrupt Mask Clear Register

Figure 14-40. INT_MSK_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		WR_MASK_CLR				LT_MASK_CLR	AT_MASK_CLR
R-0h		R/W1S-0h				R/W1S-0h	R/W1S-0h

Table 14-57. INT_MSK_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-6	RESERVED	R	0h	Reserved
5-2	WR_MASK_CLR	R/W1S	0h	Mask clear for wr_masked bits in the Interrupt Masked Register. Writing a 1 will disable the interrupts, and clear these bits as well as the wr_mask_set bits in the Interrupt Mask Set register. Writing a 0 has no effect. Reset type: SYSRSn
1	LT_MASK_CLR	R/W1S	0h	Mask clear for lt_masked bit in the Interrupt Masked Register. Writing a 1 will disable the interrupt, and clear this bit as well as the lt_mask_set bit in the Interrupt Mask Set register. Writing a 0 has no effect. Reset type: SYSRSn
0	AT_MASK_CLR	R/W1S	0h	Mask clear for at_masked bit in the Interrupt Masked Register. Writing a 1 will disable the interrupt, and clear this bit as well as the at_mask_set bit in the Interrupt Mask Set register. Writing a 0 has no effect. Reset type: SYSRSn

Chapter 15

General-Purpose Input/Output (GPIO)

The GPIO module controls the device's digital multiplexing, which uses shared pins to maximize application flexibility. The pins are named by the general-purpose I/O name (for example, GPIO0, GPIO25, GPIO58). These pins can be individually selected to operate as digital I/O (also called GPIO mode), or connected to one of several peripheral I/O signals. The input signals can be qualified to remove unwanted noise.

15.1 Introduction

Up to twelve independent peripheral signals are multiplexed on a single GPIO-enabled pin in addition to the CPU-controlled I/O capability. Each pin output can be controlled by either a peripheral or one of the CPU controllers.

There are up to 8 possible I/O ports:

- Port A consists of GPIO0-GPIO31
- Port B consists of GPIO32-GPIO63
- Port C consists of GPIO64-GPIO95
- Port D consists of GPIO96-GPIO127
- Port E consists of GPIO128-GPIO159
- Port F consists of GPIO160-GPIO191
- Port G consists of GPIO192-GPIO223
- Port H consists of GPIO224-GPIO255

Note

Some GPIO and I/O ports can be unavailable on particular devices. See the *GPIO Registers* section for available GPIO and I/O ports.

The analog signals on this device are multiplexed with digital inputs and outputs. Some of these analog IO (AIO) pins do not have digital output capability. Others of these pins are analog pins capable of full digital input and output capability (AGPIO). Analog pins with AIO (digital input only) capability contain "AIO" signals in the Pin Attributes table of the device data sheet. Analog pins with full input and output capability (AGPIO pins) contain "GPIO" signals in the Pin Attributes table of the device data sheet. AGPIO pins also have pin names with both analog signals and GPIO in the name.

Figure 15-1 shows the GPIO logic for a single pin.

There are two key features to note in Figure 15-1. The first is that the input and output paths are entirely separate, connecting only at the pin. The second is that peripheral muxing takes place far from the pin. As a result, for the CPU to read the physical state of the pin independent of peripheral muxing is possible. Likewise, external interrupts can be generated from peripheral activity. All pin options such as input qualification and open-drain output are valid for all controllers and peripherals.

Note

In open-drain mode, the GPIO does not drive the pin high, the GPIO can only pull the pin low. Instead, use an external pull-up to the bus voltage to drive the high level. When open-drain mode is enabled, the value in the GPyDAT register still controls the pin state. Writing a value of 1 turns off the driver to allow the external pull-up to control the pin; writing a value of 0 pulls the pin to ground. The open-drain configuration is automatically used by peripherals such as I2C and PMBus (no need to enable open-drain mode locally). This mode can also be set manually by writing to the GPyODR register and can be used when there are multiple nodes on the same net to avoid the pin contention that a push-pull driver can cause.

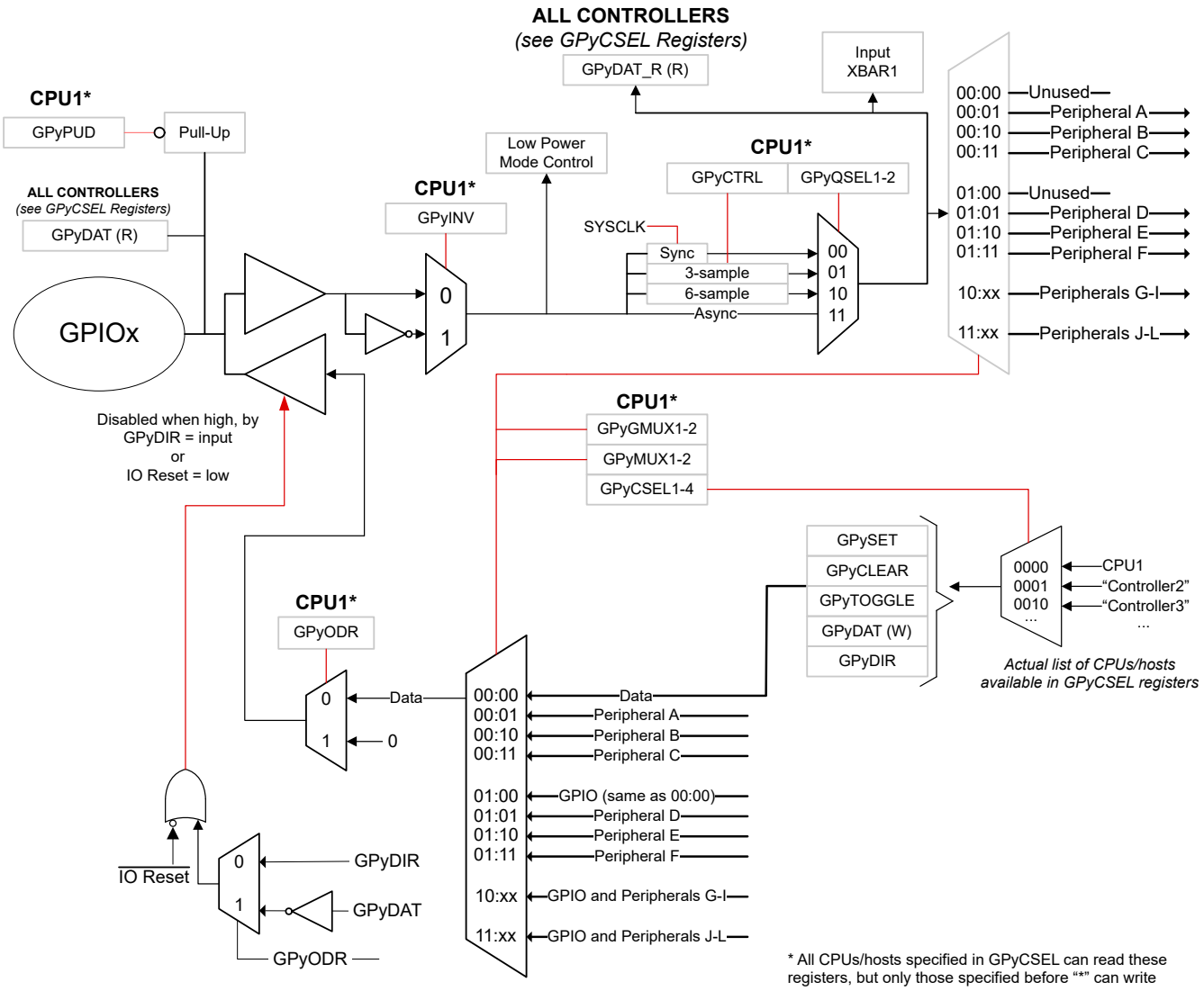


Figure 15-1. GPIO Logic for a Single Pin

15.1.1 GPIO Related Collateral

Foundational Materials

- C28x Academy - GPIO
- C29x Academy - GPIOs

Getting Started Materials

- *How to Maximize GPIO Usage in C2000 Devices Application Report*
- [FAQ] C2000 GPIO FAQ

15.2 Configuration Overview

I/O pin configuration consists of several steps:

1. **Plan the device pin-out:** Make a list of all required peripherals for the application. Using the peripheral mux information in the device data sheet, choose which GPIOs to use for the peripheral signals. Decide which of the remaining GPIOs to use as inputs and outputs for each CPU.

   Once the peripheral muxing has been chosen, implement the mux by writing the appropriate values to the GPyMUX1/2 and GPyGMUX1/2 registers. When changing the GPyGMUX value for a pin, always set the corresponding GPyMUX bits to zero first to avoid glitching in the muxes. By default, all pins are general-purpose I/Os, not peripheral signals, with the exception of GPIO35 and GPIO37.
2. **(Optional) Enable internal pullup resistors:** To enable or disable the pullup resistors, write to the appropriate bits in the GPIO pullup disable registers (GPyPUD). All pullups are disabled by default. Pullups can be used to keep input pins in a known state when there is no external signal driving them.
3. **Select input qualification:** If the pin is used as an input, specify the required input qualification, if any. The input qualification sampling period is selected in the GPyCTRL registers, while the type of qualification is selected in the GPyQSEL1 and GPyQSEL2 registers. By default, all qualification is synchronous with a sampling period equal to PLLSYSCLK, with the exception of GPIO35 and GPIO37. For an explanation of input qualification, see Section 15.6.
4. **Select the direction of any general-purpose I/O pins:** For each pin configured as a GPIO, specify the direction of the pin as either input or output using the GPyDIR registers. By default, all GPIO pins are inputs. Before changing a pin to an output, load the output latch with the value to be driven by writing that value to the GPySET, GPyCLEAR, or GPyDAT registers. Once the latch is loaded, write to GPyDIR to change the pin direction. By default, all output latches are zero.

   The GPyDAT_R register can be used to read what value was written to the GPyDAT register.
5. **Select low-power mode wake-up sources:** GPIOs 0-63 can be used to wake the system up from low power modes. To select one or more GPIOs for wake-up, write to the appropriate bits in the GPIOLPMSEL0 and GPIOLPMSEL1 registers. These registers are part of the CPU system register space. For more information on low-power modes and GPIO wake-up, see the Low-Power Modes section in the *System Control and Interrupts* chapter.
6. **Select external interrupt sources:** Configuring external interrupts is a two-step process. First, the interrupts themselves must be enabled and the polarity must be configured using the XINTnCR registers. Second, the XINT1-5 GPIO pins must be set by selecting the sources for Input X-BAR signals 4, 5, 6, 13, and 14, respectively. For more information on the Input X-BAR architecture, see the *Crossbar (X-BAR)* chapter.

15.3 Digital Inputs on ADC Pins (AIOs)

Some GPIOs are multiplexed with analog pins and only have digital input functionality. These are also referred to as AIOs. Pins with only an AIO option on this port can only function in input mode. See the device data sheet for list of AIO signals. By default, these pins function as analog pins and the GPIOs are in a high-impedance state. The GPyAMSEL register is used to configure these pins for digital or analog operation.

Note

If digital signals with sharp edges (high dv/dt) are connected to the AIOs, cross-talk can occur with adjacent analog signals. Therefore, limit the edge rate of signals connected to AIOs if adjacent channels are being used for analog functions.

15.4 Digital Inputs and Outputs on ADC Pins (AGPIOs)

Some GPIOs are multiplexed with analog pins and have digital input and output functionality. These are also referred to as AGPIOs. Unlike AIOs, AGPIOs have full input and output capability. By default, the AGPIOs are not connected and must be configured. Table 15-1 shows how to configure the AGPIOs. To enable the analog functionality, set the register AGPIOCTRLx from analog subsystem. To enable the digital functionality, set the register GPxAMSEL from the *General-Purpose Input/Output (GPIO)* chapter.

Table 15-1. AGPIO Configuration

AGPIOCTRLx.GPIOy (Default = 0)	GPxAMSEL.GPIOy (Default = 1)	Pin Connected To:	
		ADC	GPIOy
0	0	-	Yes
0	**1**	**-** (1)	**-** (1)
1	0	-	Yes
1	1	Yes	-

(1) By default there are no signals connected to AGPIO pins. One of the other rows in the table must be chosen for pin functionality.

Note

If digital signals with sharp edges (high dv/dt) are connected to the AGPIOs, cross-talk can occur with adjacent analog signals. The user must therefore limit the edge rate of signals connected to AGPIOs, if adjacent channels are being used for analog functions.

15.5 Digital General-Purpose I/O Control

The values on the pins that are configured as GPIO can be changed by using the following registers.

- **GPyDAT Registers**

  Each I/O port has one data register. Each bit in the data register corresponds to one GPIO pin. No matter how the pin is configured (GPIO or peripheral function), the corresponding bit in the data register reflects the current state of the pin after qualification. Writing to the GPyDAT register clears or sets the corresponding output latch and if the pin is enabled as a general-purpose output (GPIO output), the pin is also driven either low or high. If the pin is not configured as a GPIO output, then the value is latched but the pin is not driven. Only if the pin is later configured as a GPIO output is the latched value driven onto the pin.

  When using the GPyDAT register to change the level of an output pin, be cautious to not accidentally change the level of another pin. For example, to change the output latch level of GPIOA0 by writing to the GPADAT register bit 0 using a read-modify-write instruction, a problem can occur if another I/O port A signal changes level between the read and the write stage of the instruction. Following is an analysis of why this happens:

  The GPyDAT registers reflect the state of the pin, not the latch. This means the register reflects the actual pin value. However, there is a lag between when the register is written to when the new pin value is reflected back in the register. This can pose a problem when this register is used in subsequent program statements to alter the state of GPIO pins. An example is shown below where two program statements attempt to drive two different GPIO pins that are currently low to a high state.

  If Read-Modify-Write operations are used on the GPyDAT registers, because of the delay between the output and the input of the first instruction (I1), the second instruction (I2) reads the old value and writes the value back.

```
GpioDataRegs.GPADAT.bit.GPIO1 = 1;  //I1 performs read-modify-write of GPADAT
GpioDataRegs.GPADAT.bit.GPIO2 = 1;  //I2 also a read-modify-write of GPADAT
                                    //GPADAT gets the old value of GPIO1 due to the delay
```

  The second instruction waits for the first to finish the write due to the write-followed-by-read protection on this peripheral frame. There is some lag, however, between the write of (I1) and the GPyDAT bit reflecting the new value (1) on the pin. During this lag, the second instruction reads the old value of GPIO1 (0) and writes the value back along with the new value of GPIO2 (1). Therefore, GPIO1 pin stays low.

  One answer is to put some NOPs between instructions. A better answer is to use the GPySET/GPyCLEAR/GPyTOGGLE registers instead of the GPyDAT registers. These registers always read back a 0 and writes of 0 have no effect. Only bits that need to be changed can be specified without disturbing any other bits that are currently in the process of changing.

- **GPyDAT_R Registers**

  The GPyDAT_R registers are read only registers that return the value written to the GPyDAT registers instead of pin status. Writes to these registers have no effect.

- **GPySET Registers**

  The set registers are used to drive specified GPIO pins high without disturbing other pins. Each I/O port has one set register and each bit corresponds to one GPIO pin. The set registers always read back 0. If the corresponding pin is configured as an output, then writing a 1 to that bit in the set register sets the output latch high and the corresponding pin is driven high. If the pin is not configured as a GPIO output, then the value is latched but the pin is not driven. Only if the pin is later configured as a GPIO output is the latched value driven onto the pin. Writing a 0 to any bit in the set registers has no effect.

- **GPyCLEAR Registers**

  The clear registers are used to drive specified GPIO pins low without disturbing other pins. Each I/O port has one clear register. The clear registers always read back 0. If the corresponding pin is configured as a general-purpose output, then writing a 1 to the corresponding bit in the clear register clears the output latch and the pin is driven low. If the pin is not configured as a GPIO output, then the value is latched but the pin is not driven. Only if the pin is later configured as a GPIO output is the latched value driven onto the pin. Writing a 0 to any bit in the clear registers has no effect.

- **GPyTOGGLE Registers**

  The toggle registers are used to drive specified GPIO pins to the opposite level without disturbing other pins. Each I/O port has one toggle register. The toggle registers always read back 0. If the corresponding pin is configured as an output, then writing a 1 to that bit in the toggle register flips the output latch and pulls the corresponding pin in the opposite direction. That is, if the output pin is driven low, then writing a 1 to the corresponding bit in the toggle register pulls the pin high. Likewise, if the output pin is high, then writing a 1 to the corresponding bit in the toggle register pulls the pin low. If the pin is not configured as a GPIO output, then the value is latched but the pin is not driven. Only if the pin is later configured as a GPIO output is the latched value driven onto the pin. Writing a 0 to any bit in the toggle registers has no effect.

15.6 Input Qualification

The input qualification scheme has been designed to be very flexible. Select the type of input qualification for each GPIO pin by configuring the GPyQSEL1 and GPyQSEL2 registers. In the case of a GPIO input pin, the qualification can be specified as only synchronized to SYSCLKOUT or qualification by a sampling window. For pins that are configured as peripheral inputs, the input can also be asynchronous in addition to synchronized to SYSCLKOUT or qualified by a sampling window. The remainder of this section describes the options available.

15.6.1 No Synchronization (Asynchronous Input)

This mode is used for peripherals where input synchronization is not required or the peripheral performs the synchronization. Examples include communication ports McBSP, SCI, SPI, and I^2C. In addition, the ePWM trip zone ($\overline{TZn}$) signals can function independent of the presence of SYSCLKOUT.

Note

Using input synchronization when the peripheral performs the synchronization can cause unexpected results. The user must make sure that the GPIO pin is configured for asynchronous in this case.

15.6.2 Synchronization to SYSCLKOUT Only

This is the default qualification mode of all the pins at reset. In this mode, the input signal is only synchronized to the system clock (SYSCLKOUT). Because the incoming signal is asynchronous, a SYSCLKOUT period of delay is needed for the input to the device to be changed. No further qualification is performed on the signal.

15.6.3 Qualification Using a Sampling Window

In this mode, the signal is first synchronized to the system clock (SYSCLKOUT) and then qualified by a specified number of cycles before the input is allowed to change. Figure 15-2 and Figure 15-3 show how the input qualification is performed to eliminate unwanted noise. Two parameters are specified by the user for this type of qualification: 1) the sampling period, or how often the signal is sampled, and 2) the number of samples to be taken.

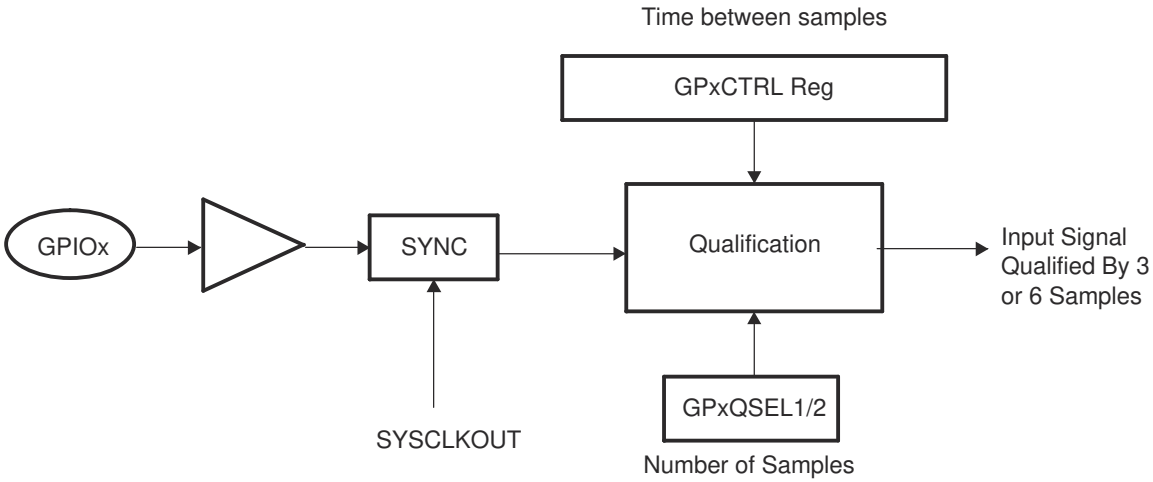


Figure 15-2. Input Qualification Using a Sampling Window

Time between samples (sampling period):

To qualify the signal, the input signal is sampled at a regular period. The sampling period is specified by the user and determines the time duration between samples, or how often the signal is sampled, relative to the CPU clock (SYSCLKOUT).

The sampling period is specified by the qualification period (QUALPRDn) bits in the GPxCTRL register . The sampling period is configurable in groups of 8 input signals. For example, GPIO0 to GPIO7 use GPACTRL[QUALPRD0] setting and GPIO8 to GPIO15 use GPACTRL[QUALPRD1]. Table 15-2 and Table 15-3 show the relationship between the sampling period or sampling frequency and the GPxCTRL[QUALPRDn] setting.

Table 15-2. Sampling Period

	Sampling Period
If GPxCTRL[QUALPRDn] = 0	1 × $T_{SYSCLKOUT}$
If GPxCTRL[QUALPRDn] ≠ 0	2 × GPxCTRL[QUALPRDn] × $T_{SYSCLKOUT}$
	Where $T_{SYSCLKOUT}$ is the period in time of SYSCLKOUT

Table 15-3. Sampling Frequency

	Sampling Frequency
If GPxCTRL[QUALPRDn] = 0	$f_{SYSCLKOUT}$
If GPxCTRL[QUALPRDn] ≠ 0	$f_{SYSCLKOUT}$ × 1 ÷ (2 × GPxCTRL[QUALPRDn])
	Where $f_{SYSCLKOUT}$ is the frequency of SYSCLKOUT

From these equations, the minimum and maximum time between samples can be calculated for a given SYSCLKOUT frequency:

Example: Maximum Sampling Frequency:

If GPxCTRL[QUALPRDn] = 0

then the sampling frequency is $f_{SYSCLKOUT}$

If, for example, $f_{SYSCLKOUT}$ = 60MHz

then the signal is sampled at 60MHz or one sample every 16.67ns.

Example: Minimum Sampling Frequency:

If GPxCTRL[QUALPRDn] = 0xFF (255)

then the sampling frequency is $f_{SYSCLKOUT} \times 1 \div (2 \times$ GPxCTRL[QUALPRDn])

If, for example, $f_{SYSCLKOUT}$ = 60MHz

then the signal is sampled at 60MHz × 1 ÷ (2 × 255) (117.647kHz) or one sample every 8.5µs.

Number of samples:

The number of times the signal is sampled is either three samples or six samples as specified in the qualification selection (GPyQSEL1, GPyQSEL2) registers. When three or six consecutive cycles are the same, then the input change is passed through to the device.

Total Sampling-Window Width:

The sampling window is the time during which the input signal is sampled as shown in Figure 15-3. By using the equation for the sampling period, along with the number of samples to be taken, the total width of the window can be determined.

For the input qualifier to detect a change in the input, the level of the signal must be stable for the duration of the sampling-window width or longer.

The number of sampling periods within the window is always one less than the number of samples taken. For a three-sample window, the sampling-window width is two sampling-periods wide where the sampling period is defined in Table 15-2. Likewise, for a six-sample window, the sampling-window width is five sampling-periods wide. Table 15-4 and Table 15-5 show the calculations used to determine the total sampling-window width based on GPxCTRL[QUALPRDn] and the number of samples taken.

Table 15-4. Case 1: Three-Sample Sampling-Window Width

	Total Sampling-Window Width
If GPxCTRL[QUALPRDn] = 0	2 × $T_{SYSCLKOUT}$
If GPxCTRL[QUALPRDn] ≠ 0	2 × 2 × GPxCTRL[QUALPRDn] × $T_{SYSCLKOUT}$
	Where $T_{SYSCLKOUT}$ is the period in time of SYSCLKOUT

Table 15-5. Case 2: Six-Sample Sampling-Window Width

	Total Sampling-Window Width
If GPxCTRL[QUALPRDn] = 0	5 × $T_{SYSCLKOUT}$
If GPxCTRL[QUALPRDn] ≠ 0	5 × 2 × GPxCTRL[QUALPRDn] × $T_{SYSCLKOUT}$
	Where $T_{SYSCLKOUT}$ is the period in time of SYSCLKOUT

Note

The external signal change is asynchronous with respect to both the sampling period and SYSCLKOUT. Due to the asynchronous nature of the external signal, the input must be held stable for a time greater than the sampling-window width to make sure the logic detects a change in the signal. The extra time required can be up to an additional sampling period + $T_{SYSCLKOUT}$.

The required duration for an input signal to be stable for the qualification logic to detect a change is described in the data sheet.

Example Qualification Window:

For the example shown in Figure 15-3, the input qualification has been configured as follows:

- GPxQSEL1/2 = 1,0. This indicates a six-sample qualification.
- GPxCTRL[QUALPRDn] = 1. The sampling period is $t_w(SP) = 2 \times$ GPxCTRL[QUALPRDn] $\times T_{SYSCLKOUT} = 2 \times T_{SYSCLKOUT}$.

This configuration results in the following:

- The width of the sampling window is:

  $t_w(IQSW) = 5 \times t_w(SP) = 5 \times 2 \times$ GPxCTRL[QUALPRDn] $\times T_{SYSCLKOUT} = 5 \times 2 \times T_{SYSCLKOUT}$

- If, for example, $T_{SYSCLKOUT}$ = 16.67ns, then the duration of the sampling window is:

  Sampling period, $t_w(SP) = 2 \times T_{SYSCLKOUT} = 2 \times 16.67ns = 33.3ns$

  Sampling window, $t_w(IQSW) = 5 \times t_w(SP) = 5 \times 33.3ns = 166.7ns$

- To account for the asynchronous nature of the input relative to the sampling period and SYSCLKOUT, up to a single additional sampling period and SYSCLK period is required to detect a change in the input signal. For this example:

  $t_w(IQSW) + t_w(SP) + T_{SYSCLKOUT} = 166.7ns + 33.3ns + 16.67ns = 216.7ns$

- In Figure 15-3, the glitch (A) is shorter then the qualification window and is ignored by the input qualifier.

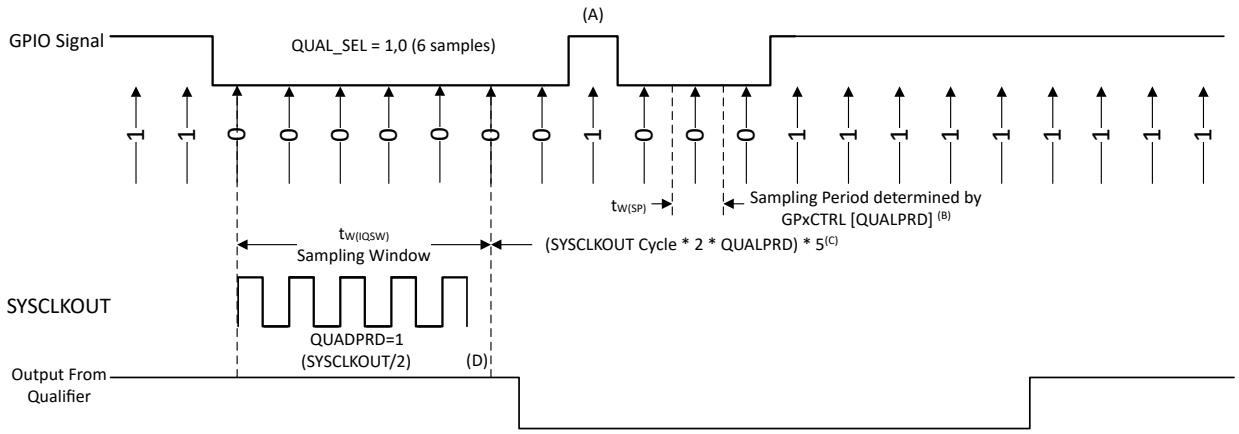


Figure 15-3. Input Qualifier Clock Cycles

- **A.** This glitch will be ignored by the input qualifier. The QUALPRD bit field specifies the qualification sampling period. It can vary from 0x00 to 0xFF. If QUALPRD = 00, then the sampling period is 1 SYSCLKOUT cycle. For any other value 'n', the qualificiation sampling period is 2n SYSCLKOUT cycles (i.e. at every 2n SYSCLKOUT cycles, the GPIO pin will be sampled).
- **B.** The qualification period selected via the GPxCTRL register applies to groups of 8 GPIO pins.

- **C.** the qualification block can take either 3 or 6 samples. The QUAL_SEL Register selects which samples mode is used.
- **D.** In the example shown, for the qualifier to detect the change, the input should be stable for 10 SYSCLKOUT cycles. That would ensure 5 sampling periods for detection to occur. Since external signals are driven asynchronously, a 13-SYSCLKOUT-wide pulse ensures reliable recognition.

15.7 PMBUS and I2C Signals

To support a wider range of PMBUS and I2C IO levels, certain GPIOs on this device have configurable V_{IH} minimum thresholds and configurable current sinking capabilities.

- PMBUS_IO_MODESEL register configures the V_{IH} threshold of the GPIO
- PMBUS_IO_DRVSEL register configures the current sinking capability of the GPIO

Note

The PMBUS_IO_MODESEL and PMBUS_IO_DRVSEL registers apply to the entire GPIO, not just the PMBUS module. Any peripheral or module in the given GPIO's mux is able to utilize the customizable V_{IH} threshold and current sinking capability.

The list of GPIOs that have these capabilities, and the configurable levels for these GPIOs are available in the PMBUS_IO_MODESEL and PMBUS_IO_DRVSEL registers.

15.8 GPIO and Peripheral Muxing

15.8.1 GPIO Muxing

Table 15-6. GPIO Muxed Pins

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO0	EPWM1_A	EMIF1_A13	EMIF1_D0	MCAND_TX	I2CA_SDA	UARTE_TX	OUTPUTXBAR9	ESC_TX0_DATA0	ESC_GPI0	FSITXA_D0			
GPIO1	EPWM1_B	EMIF1_A14	EMIF1_D3	MCAND_RX	I2CA_SCL	UARTE_RX	OUTPUTXBAR10	ESC_TX1_DATA0	ESC_GPI1	FSITXA_D1			
GPIO2	EPWM2_A	EMIF1_A15	EMIF1_D4	UARTA_TX	I2CB_SDA	MCANF_TX	OUTPUTXBAR1	ESC_RX1_ERR	ESC_GPI2	FSITXA_CLK			
GPIO3	EPWM2_B	EMIF1_A16	EMIF1_D5	UARTA_RX	I2CB_SCL	MCANF_RX	OUTPUTXBAR2		ESC_GPI3	FSIRXA_D0			
GPIO4	EPWM3_A	EMIF1_A17	EMIF1_D9	MCANC_TX		UARTF_TX	OUTPUTXBAR3		ESC_GPI4	FSIRXA_D1		ERRORSTS	
GPIO5	EPWM3_B	EMIF1_A18	EMIF1_D10	MCANC_RX		UARTF_RX	OUTPUTXBAR11	OUTPUTXBAR3	ESC_GPI5	FSIRXA_CLK			
GPIO6	EPWM4_A	EMIF1_DQM0	EMIF1_CLK	MCANB_TX	LINA_TX		OUTPUTXBAR4	SYNCOUT	ESC_GPI6	FSITXB_D0			
GPIO7	EPWM4_B	EMIF1_DQM1	EMIF1_CAS	MCANB_RX	LINA_RX		OUTPUTXBAR5		ESC_GPI7	FSITXB_D1			
GPIO8	EPWM5_A	EMIF1_RAS	EPWM4_B	MCANC_TX	SPIE_PICO	UARTD_TX	OUTPUTXBAR12	ADCSOCAO	ESC_GPO0	FSITXB_CLK	FSITXA_D1	FSIRXA_D0	
GPIO9	EPWM5_B	EMIF1_D11			SPIE_POCI	UARTD_RX	OUTPUTXBAR6	ESC_TX0_CLK	ESC_GPO1	FSIRXB_D0	FSITXA_D0	FSIRXA_CLK	
GPIO10	EPWM8_A	PMBUSA_SCL	ADCSOCBO	MCANC_RX	UARTC_TX	I2CA_SCL	SENT2			ESC_GPI19	ADCA_EXTMUXSEL2	OUTPUTXBAR13	
GPIO11	EPWM6_B	EMIF1_D15	EPWM7_B		SPIE_PTE	SD4_D1	PMBUSA_ALERT	ESC_TX0_DATA1	ESC_GPO3	FSIRXB_CLK	FSIRXA_D1	OUTPUTXBAR7	
GPIO12	EPWM7_A	EMIF1_A1	ADCSOCAO		SPIE_CLK	SD4_C2	PMBUSA_CTL	ESC_TX0_DATA2	ESC_GPO4	FSIRXC_D0	FSIRXA_D0	OUTPUTXBAR14	
GPIO13	EPWM7_B	EMIF1_CS0n	EMIF1_D9		UARTC_RX	SD4_D2	PMBUSA_SDA	ESC_TX0_DATA3	ESC_GPO5	FSIRXC_D1	FSIRXA_CLK	OUTPUTXBAR15	
GPIO14	EPWM6_A	EMIF1_D17	EPWM18_A	EMIF1_D13	LINA_TX	OUTPUTXBAR3	PMBUSA_SCL	ESC_PHY1_LINKSTATUS	ESC_GPO6	FSIRXC_CLK	SD4_C1	OUTPUTXBAR8	
GPIO15	EPWM8_B		PMBUSA_CTL	I2CA_SDA	LINA_RX	OUTPUTXBAR4	SENT1	ESC_GPO7		ESC_GPI20	ADCA_EXTMUXSEL3	OUTPUTXBAR16	
GPIO16	EPWM9_A	EMIF1_D29	EMIF1_BA0	SPIA_PICO		MCAND_TX		ESC_RX1_CLK	SD1_D1	FSIRXD_D1	FSIRXC_CLK	OUTPUTXBAR7	
GPIO17	EPWM9_B	EMIF1_DQM3	EMIF1_BA1	SPIA_POCI		MCAND_RX		ESC_RX1_DV	SD1_C1	FSIRXD_CLK	UARTC_TX	OUTPUTXBAR8	

Table 15-6. GPIO Muxed Pins (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO18	EPWM15_A		PMBUSA_ALERT	I2CA_SCL	UARTC_RX		SENT4			ESC_GPI21	ADCB_EXTMUXSEL0		
GPIO19	EPWM10_B	EMIF1_CS3n	ADCSOCBO	SPIA_PTE	UARTE_RX	MCANC_TX	PMBUSA_ALERT	ESC_TX1_DATA3	SD1_C2				
GPIO20	EPWM11_A	EMIF1_BA0	EMIF1_DQM2		SPIC_PICO	MCANB_RX		ESC_TX1_DATA2	SD1_D3				
GPIO21	EPWM11_B	EMIF1_BA1			SPIC_POCI	MCANB_TX		ESC_TX1_DATA1	SD1_C3				
GPIO22	EPWM12_A		PMBUSA_SDA	I2CB_SDA	UARTB_TX	MCANC_TX	SENT5	ESC_GPO2		ESC_GPI22	ADCB_EXTMUXSEL1		
GPIO23	EPWM12_B		PMBUSA_SCL	I2CB_SCL	UARTB_RX	MCANC_RX	SENT6	ESC_PHY_RESETn		ESC_GPI23	ADCC_EXTMUXSEL0		
GPIO24	EPWM13_A	EMIF1_DQM0		SPIB_PICO	LINB_TX	MCANE_TX		ESC_RX0_CLK	SD2_D1	ESC_GPI24	EPWM2_A	OUTPUTXBAR1	
GPIO25	EPWM13_B	EMIF1_DQM1		SPIB_POCI	LINB_RX	MCANE_RX	PMBUSA_SDA	ESC_RX0_DV	SD2_C1	FSITXA_D1	EPWM2_B	OUTPUTXBAR2	
GPIO26	EPWM14_A	EMIF1_DQM2		SPIB_CLK	UARTE_TX	MCANE_TX	PMBUSA_CTL	ESC_RX0_ERR	SD2_D2	FSITXA_D0	ESC_MDIO_CLK	OUTPUTXBAR3	
GPIO27	EPWM14_B	EMIF1_DQM3		SPIB_PTE	UARTA_TX		EPWM4_A	ESC_RX0_DATA0	SD2_C2	FSITXA_CLK	ESC_MDIO_DATA	OUTPUTXBAR4	
GPIO28	EPWM15_A	EMIF1_CS4n	EMIF1_CS2n		UARTA_RX		EPWM4_B	ESC_RX0_DATA1	SD2_D3			OUTPUTXBAR5	
GPIO29	EPWM15_B	PMBUSA_SDA			UARTE_RX	I2CA_SDA	SENT3	ESC_LATCH0		ESC_I2C_SDA	ADCC_EXTMUXSEL1	OUTPUTXBAR6	
GPIO30	EPWM16_A	EMIF1_CLK	EMIF1_CS4n	MCANC_RX	SPID_PICO	EMIF1_A12		ESC_LATCH1	SD2_D4	ESC_I2C_SCL	ESC_SYNC1	OUTPUTXBAR7	
GPIO31	EPWM16_B	EMIF1_WEn	EMIF1_RNW	MCANC_TX	SPID_POCI	I2CA_SDA		ESC_RX1_DATA0	SD2_C4	FSITXD_D0		OUTPUTXBAR8	
GPIO32		EMIF1_CS0n	EMIF1_OEn	SPIA_PICO	SPID_CLK	I2CA_SDA	OUTPUTXBAR9	ESC_RX0_DATA0					
GPIO33		EMIF1_RNW	EMIF1_BA0	SPIA_POCI	SPID_PTE	I2CA_SCL	OUTPUTXBAR10	ESC_LED_ERR					

Table 15-6. GPIO Muxed Pins (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO34	EPWM18_A	EMIF1_CS2n	EMIF1_BA1	SPIA_CLK	UARTF_TX	I2CB_SDA	OUTPUTXBAR11	ESC_LATCH0		EPWM3_B	ESC_SYNC0	OUTPUTXBAR1	
GPIO35	EPWM18_B	EMIF1_CS3n	EMIF1_A0	SPIA_PTE	UARTF_RX	I2CB_SCL	OUTPUTXBAR12	ESC_LATCH1			ESC_SYNC1		
GPIO36		EMIF1_WAIT	EMIF1_A1	UARTC_TX	MCANC_RX		OUTPUTXBAR13		SD1_D1		EMIF1_WEn		
GPIO37	EPWM18_A	EMIF1_OEn	EMIF1_A2	UARTC_RX	MCANC_TX		OUTPUTXBAR14	ESC_RX1_DATA1	SD1_D2		EMIF1_D24	OUTPUTXBAR2	
GPIO38	EPWM18_B	EMIF1_A0	EMIF1_A3	UARTA_TX	SPIE_PICO		OUTPUTXBAR15	ESC_RX0_DATA1	SD1_D3	FSITXD_D1	EMIF1_CS2n		
GPIO39		EMIF1_A1	EMIF1_A4	UARTA_RX			OUTPUTXBAR16	ESC_MDIO_DATA	SD1_D4	FSIRXD_CLK		ESC_LED_RUN	
GPIO40	EPWM13_A	EMIF1_A2		MCANB_RX	I2CB_SDA		OUTPUTXBAR9	ESC_GPO2	SD4_C3		EPWM1_A	SD2_C1	
GPIO41	EPWM13_B	EMIF1_A3	EPWM18_A	MCANB_TX	SPIE_POCI	I2CB_SCL	OUTPUTXBAR10	ESC_RX0_DATA2	SD4_D3	FSIRXD_CLK	EPWM1_B	SD2_D1	
GPIO42	EPWM14_A	EMIF1_A2	EMIF1_A13	UARTA_TX	SPIE_CLK	I2CA_SDA	OUTPUTXBAR13	SD4_C3	SD4_C4	FSIRXD_D0	ADCE_EXTMUXSEL2		
GPIO43	EPWM14_B	EMIF1_A4	EMIF1_D13	UARTA_RX	SPIE_PTE	I2CA_SCL	OUTPUTXBAR14		SD4_D4	FSIRXD_D1	ADCE_EXTMUXSEL3		
GPIO44		EMIF1_A4		SPID_POCI	MCANB_RX	UARTB_TX	OUTPUTXBAR14	ESC_TX1_CLK	SD3_C4	FSIRXD_CLK			
GPIO45		EMIF1_A5		SPID_PTE	MCANB_TX	UARTB_RX	OUTPUTXBAR15	ESC_TX1_ENA	SD3_D4	FSIRXD_D0			
GPIO46	EPWM4_A	EMIF1_A6	EPWM14_A	UARTC_TX		MCANE_TX		ESC_MDIO_CLK	SD3_C4				
GPIO47	EPWM4_B	EMIF1_A7	EPWM14_B	UARTC_RX		MCANE_RX		ESC_MDIO_DATA	SD4_C3				
GPIO48		EMIF1_A8		UARTD_TX			OUTPUTXBAR3	ESC_PHY_CLK	SD1_D1	EPWM3_A		SD2_C2	
GPIO49		EMIF1_A9	EMIF1_A5	UARTD_RX			OUTPUTXBAR4	ESC_TX1_DATA2	SD1_C1	FSITXA_D0		SD2_D1	
GPIO50	EPWM15_A	EMIF1_A10	EMIF1_A6		SPIC_PICO	MCANF_TX		ESC_TX1_DATA1	SD1_D2	FSITXA_D1	ESC_GPI25	SD2_D2	

Table 15-6. GPIO Muxed Pins (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO51	EPWM15_B	EMIF1_A11	EMIF1_A7		SPIC_POCI	MCANF_RX		ESC_TX1_CLK	SD1_C2	FSITXA_CLK	ESC_GPI26	SD2_D3	
GPIO52	EPWM16_A	EMIF1_A12	EMIF1_A8	UARTD_TX	SPIC_CLK			ESC_TX1_ENA	SD1_D3	FSIRXA_D0		SD2_D4	
GPIO53	EPWM16_B	EMIF1_D31	EMIF1_A9	UARTD_RX	SPIC_PTE			ESC_PHY0_LINKSTATUS	SD1_C3	FSIRXA_D1	ESC_GPI28	SD1_C1	
GPIO54		EMIF1_D30	EMIF1_A10	SPIA_PICO				ESC_PHY_CLK	SD1_D4	FSIRXA_CLK	ESC_GPI29	SD1_C2	
GPIO55	EPWM16_B	EMIF1_D29	EMIF1_D0	SPIA_POCI	EMIF1_WAIT			ESC_PHY0_LINKSTATUS	SD1_C4	FSITXB_D0		SD1_C3	
GPIO56	EPWM17_A	EMIF1_D28	EMIF1_D1	SPIA_CLK	MCAND_TX	I2CA_SDA		ESC_PDI_UC_IRQ	SD2_D1	FSITXB_CLK	ESC_GPI30	SD1_C4	
GPIO57	EPWM17_B	EMIF1_D27	EMIF1_D2	SPIA_PTE	MCAND_RX	I2CA_SCL		ESC_MDIO_DATA	SD2_C1	FSITXB_D1	ESC_GPI31	SD3_D3	
GPIO58	EPWM13_A	EMIF1_D26	EPWM8_A	SPIA_PICO		MCANC_RX	SENT1	ESC_LED_LINK0_ACTIVE	SD2_D2	FSIRXB_D0	ESC_TX0_DATA3	SD2_C2	
GPIO59	EPWM5_A	EMIF1_D25	EPWM8_B	SPIA_POCI		MCANC_TX	SENT2	ESC_LED_LINK1_ACTIVE	SD2_C2	FSIRXB_D1	ESC_TX0_ENA	SD2_C3	
GPIO60	EPWM3_B	EMIF1_D24	EMIF1_D0	SPIA_CLK	OUTPUTXBAR3		SENT3	ESC_LED_ERR	ESC_LATCH0	FSIRXB_CLK		SD2_C4	
GPIO61	EPWM17_B	EMIF1_D23	EMIF1_D6	SPIA_PTE		MCANC_RX	OUTPUTXBAR4	ESC_LED_RUN	SD2_C3	FSITXD_CLK	ESC_LATCH1		
GPIO62	EPWM17_A	EMIF1_D22	EMIF1_D7		MCANC_RX	MCANC_TX	SENT4	ESC_LED_STATE_RUN	SD2_D4	FSITXD_D0	ESC_MDIO_CLK		
GPIO63	EPWM9_A	EMIF1_D21	EMIF1_RNW	SPIB_PICO	MCANC_TX		SENT5	ESC_RX1_DATA0	SD1_D1	FSITXD_D1	ADCD_EXTMUXSEL0	SD2_C4	
GPIO64	EPWM9_B	EMIF1_D20	EMIF1_WAIT	SPIB_POCI	MCANA_TX	UARTF_TX	SENT6	ESC_RX1_DATA1	SD1_C1	FSITXD_CLK	ADCD_EXTMUXSEL1		
GPIO65	EPWM10_A	EMIF1_D19	EMIF1_WEn	SPIB_CLK	MCANA_RX	UARTF_RX		ESC_RX1_DATA2	SD1_D2	FSITXB_CLK	ADCD_EXTMUXSEL2	ESC_GPI13	
GPIO66	EPWM10_B	EMIF1_D18	EMIF1_OEn	SPIB_PTE	I2CB_SDA			ESC_RX1_DATA3	SD1_C2	FSITXB_D1	ADCD_EXTMUXSEL3	ESC_GPI14	

Table 15-6. GPIO Muxed Pins (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO67	EPWM17_A	EMIF1_D17		LINB_TX	MCAND_TX				SD1_D3	FSITXB_CLK			
GPIO68	EPWM17_B	EMIF1_D16	EMIF1_D4	LINB_RX	MCAND_RX	EMIF1_D13		ESC_PHY1_LINKSTATUS	SD1_C3	FSIRXB_D1		ESC_GPI15	
GPIO69	EPWM11_A	EMIF1_D15		SPIC_PICO	I2CB_SCL			ESC_RX1_CLK	SD1_D4	FSITXB_D0			
GPIO70	EPWM11_B	EMIF1_D14		SPIC_POCI	MCANC_RX	UARTB_TX		ESC_RX1_DV	SD1_C4	FSIRXB_D0		ESC_GPI16	
GPIO71	EPWM12_A	EPWM11_A	EMIF1_D5	SPIC_CLK	MCANC_TX	UARTB_RX	EMIF1_D13	ESC_RX1_ERR	SD3_D1	FSITXC_CLK	FSITXB_D0		
GPIO72	EPWM12_B	EMIF1_D12		SPIC_PTE	MCANB_RX	UARTA_TX	OUTPUTXBAR8	ESC_TX1_DATA3	SD3_D2	FSITXC_D0	SD3_C1		
GPIO73	EPWM5_B	EMIF1_D11	XCLKOUT		MCANB_TX	UARTA_RX	OUTPUTXBAR6	ESC_TX1_DATA2	SD4_D4	FSITXC_CLK	SD2_D2		
GPIO74	EPWM8_A	EMIF1_D10			MCANC_TX			ESC_TX1_DATA1	SD1_D4	FSITXA_D0	SD2_C2		
GPIO75	EPWM8_B	EMIF1_D9		SPID_CLK	MCANC_RX		OUTPUTXBAR16	ESC_TX1_DATA0			SD2_D3		
GPIO76	EPWM9_A	EMIF1_D8		UARTD_TX		MCANE_TX	SD4_D4	ESC_PHY_RESETn	SD3_C1	FSIRXC_D0	SD2_C3	ESC_GPI17	
GPIO77	EPWM9_B	EMIF1_D7		UARTD_RX		MCANE_RX	SD1_D4	ESC_RX0_CLK	SD3_D1	FSITXB_D0	SD2_D4		
GPIO78	EPWM10_A	EMIF1_D6	EPWM11_A			MCANF_TX	SD4_D4	ESC_RX0_DV	SD3_C2	FSITXC_D1	SD2_C4	ESC_GPI18	
GPIO79	EPWM10_B	EMIF1_D5		ERRORSTS				ESC_RX0_ERR	SD3_D2	FSITXC_D0	SD2_D1		
GPIO80	EPWM11_A	EMIF1_D4		ERRORSTS			SD1_D4	ESC_RX0_DATA0	SD3_C3		SD2_C1		
GPIO81	EPWM11_B	EMIF1_D3						ESC_RX0_DATA1	SD3_D3				
GPIO82	EPWM12_A	EMIF1_D2						ESC_RX0_DATA2	SD3_C2				
GPIO83	EPWM12_B	EMIF1_D1						ESC_RX0_DATA3	SD3_D2				

Table 15-6. GPIO Muxed Pins (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO84	EPWM12_B	EMIF1_D1	EMIF1_CS4n	SPIC_PICO	UARTA_TX	MCANF_RX		ESC_TX0_ENA	SD3_C2	FSITXC_D1	ESC_RX0_DATA3	ESC_GPO24	
GPIO85	EPWM13_A	EMIF1_D0			UARTA_RX		EMIF1_DQM2	ESC_TX0_CLK	SD3_D3				
GPIO86	EPWM13_B	EMIF1_A13	EMIF1_CAS		UARTD_TX			ESC_PHY0_LINKSTATUS	SD3_C3				
GPIO87	EPWM14_A	EMIF1_A14	EMIF1_RAS		UARTD_RX		EMIF1_DQM3	ESC_TX0_DATA0	SD3_D4				
GPIO88	EPWM14_B	EMIF1_A15	EMIF1_DQM0				EMIF1_DQM1	ESC_TX0_DATA1	SD3_C4				
GPIO89	EPWM15_A	EMIF1_A16	EMIF1_DQM1	SPID_PTE			EMIF1_CAS	ESC_TX0_DATA2	SD1_D3		SD4_D1		
GPIO90	EPWM15_B	EMIF1_A17	EMIF1_DQM2	SPID_CLK			EMIF1_RAS	ESC_TX0_DATA3	SD1_C3		SD4_C1		
GPIO91	EPWM16_A	EMIF1_A18	EMIF1_DQM3	SPID_PICO	I2CA_SDA	MCAND_TX	EMIF1_DQM2		SD4_D2		OUTPUTXBAR9		
GPIO92	EPWM16_B	EMIF1_A19	EMIF1_BA1	SPID_POCI	I2CA_SCL	MCAND_RX	EMIF1_DQM0	FSIRXD_CLK	SD4_C2		OUTPUTXBAR10		
GPIO93	EPWM17_A		EMIF1_BA0	SPID_CLK				ESC_TX1_CLK	SD4_D3		OUTPUTXBAR11		
GPIO94	EPWM17_B			SPID_PTE			EMIF1_BA1	ESC_TX1_ENA	SD4_C3		OUTPUTXBAR12		
GPIO95	EPWM18_A							ESC_GPO10	SD1_D1		OUTPUTXBAR13		
GPIO96	EPWM18_B							ESC_GPO11	SD1_C1		OUTPUTXBAR14		
GPIO97								ESC_GPI17	SD1_D2		OUTPUTXBAR15		
GPIO98								ESC_GPI18	SD1_C2		OUTPUTXBAR16		
GPIO99	EPWM8_A	EMIF1_DQM3	EMIF1_D17					ESC_GPI21	SD4_D4				
GPIO100	EPWM9_A	EMIF1_BA1	EMIF1_D24	SPIC_PICO	SPIA_PICO		SD1_D1	ESC_GPI0	SD4_C4	FSITXA_D0	FSIRXD_D1		

Table 15-6. GPIO Muxed Pins (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO101	EPWM18_A	EMIF1_A5		SPIC_POCI				ESC_GPI1		FSITXA_D1			
GPIO103	EPWM8_B	EMIF1_BA0	EMIF1_D3	SPIC_PTE				ESC_GPI3	SD4_C4	FSIRXA_D0		ESC_GPO25	
GPIO105	EPWM18_B			I2CA_SCL				ESC_GPI5	SD3_C1	FSIRXA_CLK			
GPIO127	EPWM18_A	EMIF1_D18	EMIF1_A11	SPID_POCI				ESC_GPI27	SD1_C3	FSIRXC_D1	ESC_SYNC0	ESC_GPO26	
GPIO219	ERRORSTS	EMIF1_A19	EPWM18_B				OUTPUTXBAR1	XCLKOUT	SD2_C1	ESC_GPI8	ESC_TX0_ENA	ESC_GPO27	
GPIO220	EPWM6_A			SPID_POCI	MCANC_TX		OUTPUTXBAR2		SD3_D3	ESC_GPI9		ESC_GPO28	X1
GPIO221	EPWM6_B		EMIF1_CAS	SPID_PTE	MCANC_RX		OUTPUTXBAR3		SD3_C3	ESC_GPI10		ESC_GPO29	X2
GPIO222	TDI	EPWM7_A		SPID_PICO	UARTB_TX	I2CB_SCL	OUTPUTXBAR4	SPIC_CLK	SD3_D4	ESC_GPI11		ESC_GPO30	
GPIO223	TDO	EPWM7_B		SPID_CLK	UARTB_RX	I2CB_SDA	OUTPUTXBAR5	SPIC_PTE	SD3_C4	ESC_GPI12		ESC_GPO31	
GPIO224	EPWM12_A	EPWM12_B		SPIB_POCI	MCAND_RX		OUTPUTXBAR5		SD4_D2		ADCA_EXTMUXSEL0	ESC_GPO8	
GPIO225	EPWM11_B			SPIB_PICO	I2CB_SDA	UARTF_TX	OUTPUTXBAR4		SD4_C1		ADCA_EXTMUXSEL1	ESC_GPO9	
GPIO226	EPWM10_A			SPIA_PTE	MCAND_TX	UARTF_RX	OUTPUTXBAR1	SD1_C3	SD1_D3		ADCA_EXTMUXSEL2	ESC_GPO10	
GPIO227	EPWM14_B			SPIA_CLK			OUTPUTXBAR4		SD2_C2		ADCA_EXTMUXSEL3		
GPIO228	EPWM18_A	EPWM13_A		SPIB_POCI	LINB_TX		OUTPUTXBAR1	SENT4	SD2_D1				
GPIO229	EPWM17_B	EPWM12_B		SPIB_PICO	MCANA_RX			SENT3	SD1_C4				
GPIO230	EPWM11_A		SYNCOUT		I2CB_SCL		OUTPUTXBAR3		SD4_D1		ADCB_EXTMUXSEL0		
GPIO231	EPWM10_B			SPIA_PICO	MCAND_RX		OUTPUTXBAR2		SD1_C3		ADCB_EXTMUXSEL1		
GPIO232	EPWM14_A	EPWM8_B		SPIA_POCI			OUTPUTXBAR3	SENT6	SD3_D1	ESC_PHY0_LINKSTATUS	ADCB_EXTMUXSEL2	ESC_GPO11	

Table 15-6. GPIO Muxed Pins (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO233	EPWM18_B	EPWM13_B			LINB_RX		OUTPUTXBAR2	SENT5	SD2_C1	ESC_PHY1_LINKSTATUS	ADCB_EXTMUXSEL 3	ESC_GPO12	
GPIO234	EPWM17_A	EPWM12_A		SPIB_PTE	MCANA_TX			SENT2	SD1_D4			ESC_GPO13	
GPIO235	EPWM9_B			SPIB_CLK	MCANA_RX			SENT1	SD1_C1			ESC_GPO14	
GPIO236	EPWM12_B	EPWM8_A			LINA_RX		OUTPUTXBAR6		SD4_C2	ESC_I2C_SDA	ADCC_EXTMUXSEL 0		
GPIO237	EPWM14_A	EPWM8_B	EPWM17_B		LINA_TX	I2CA_SDA	OUTPUTXBAR7		SD4_D3	ESC_I2C_SCL	ADCC_EXTMUXSEL 1		
GPIO238	EPWM15_B						OUTPUTXBAR6	SD1_D3	SD2_C3	ESC_SYNC0	ADCC_EXTMUXSEL 2	ESC_GPO15	
GPIO239	EPWM16_B				LINB_TX	I2CA_SCL	OUTPUTXBAR8		SD2_C4	ESC_SYNC1	ADCC_EXTMUXSEL 3	ESC_GPO16	
GPIO240	EPWM14_B			SPID_PICO					SD4_C3	ESC_LED_RUN	ADCD_EXTMUXSEL 0		
GPIO241	EPWM8_A			SPID_CLK					SD4_D4	ESC_LED_ERR	ADCD_EXTMUXSEL 1	ESC_GPO17	
GPIO242					SD1_D4	I2CA_SDA	OUTPUTXBAR9	SENT1	SD2_D2	ESC_LED_STATE_RUN	ADCD_EXTMUXSEL 2	ESC_GPO18	
GPIO243	EPWM8_B							SENT2	SD2_D4	ESC_LED_LINK0_ACTIVE	ADCD_EXTMUXSEL 3	ESC_GPO19	
GPIO244				SPIC_PTE				SENT5	SD4_C4	ESC_LED_LINK1_ACTIVE			
GPIO245				SPIC_POCI				SENT6	SD3_C1	ESC_PHY_RESETn			
GPIO246	EPWM16_A			SPID_PTE	MCANC_RX		OUTPUTXBAR7		SD1_D1		ADCE_EXTMUXSEL 0	ESC_GPO20	
GPIO247	EPWM15_A	ERRORSTS		SPID_POCI	MCANC_RX	LINA_TX	OUTPUTXBAR5		SD2_D3		ADCE_EXTMUXSEL 1	ESC_GPO21	
GPIO248		EMIF1_SDCKE		SPIC_PICO				SENT3	SD1_C2	ESC_LED_RUN	ADCE_EXTMUXSEL 2	ESC_GPO22	
GPIO249				SPIC_CLK				SENT4	SD1_D2	ESC_PHY0_LINKSTATUS	ADCE_EXTMUXSEL 3	ESC_GPO23	

Table 15-6. GPIO Muxed Pins (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
AIO160									SD3_C2				
AIO161									SD3_D2				
AIO162									SD2_C2				
AIO163									SD2_D2				
AIO164									SD2_C3				
AIO165									SD2_D3				
AIO166									SD4_C1				
AIO167									SD4_D1				
AIO168									SD3_C3				
AIO169									SD3_D3				
AIO170									SD3_C4				
AIO171									SD3_D4				
AIO172									SD1_C1				
AIO173									SD1_D1				
AIO174									SD2_C4				
AIO175									SD2_D4				
AIO176									SD4_C2				
AIO177									SD4_D2				
AIO178									SD4_C3				
AIO179									SD4_D3				
AIO180									SD1_C2				
AIO181									SD1_D2				
AIO182									SD3_C1				
AIO183									SD3_D1				
AIO184									SD3_C2				
AIO185									SD3_D2				
AIO186									SD1_C1				

Table 15-6. GPIO Muxed Pins (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
AIO187									SD1_D1				
AIO188									SD1_C2				
AIO189									SD1_D2				
AIO190									SD1_C3				
AIO191									SD1_D3				
AIO192									SD1_C3				
AIO193									SD1_D3				
AIO194									SD1_C4				
AIO195									SD1_D4				
AIO196									SD4_C4				
AIO197									SD4_D4				
AIO198									SD1_C4				
AIO199									SD1_D4				
AIO200									SD2_C1				
AIO201									SD2_D1				
AIO202									SD2_C1				
AIO203									SD2_D1				
AIO204									SD3_C3				
AIO205									SD3_D3				
AIO206									SD3_C4				
AIO207									SD3_D4				
AIO208									SD2_C2				
AIO209									SD2_D2				
AIO210									SD2_C3				
AIO211									SD2_D3				
AIO212									SD2_C4				
AIO213									SD2_D4				

15.8.2 Peripheral Muxing

For example, multiplexing for the GPIO6 pin is controlled by writing to GPAGMUX[13:12] and GPAMUX[13:12]. By writing to these bits, GPIO6 is configured as either a general-purpose digital I/O or one of several different peripheral functions. An example of GPyGMUX and GPyMUX selection and options for a single GPIO are shown in Table 15-7.

Note

The following table is for example only. Refer to the device data sheet to check the availability of GPIO6 on this device. If GPIO6 is available, the functions mentioned in the table may not match the actual functions available. See Section 15.8.1 for correct list of GPIOs and corresponding mux options for this device.

Table 15-7. GPIO and Peripheral Muxing

GPAGMUX1[13:12]	GPAMUX1[13:12]	Pin Functionality
00	00	GPIO6
00	01	Peripheral 1
00	10	Peripheral 2
00	11	Peripheral 3
01	00	GPIO6
01	01	Peripheral 4
01	10	Peripheral 5
01	11	
10	00	GPIO6
10	01	
10	10	Peripheral 6
10	11	Peripheral 7
11	00	GPIO6
11	01	Peripheral 8
11	10	Peripheral 9
11	11	Peripheral 10

The devices have different multiplexing schemes. If a peripheral is not available on a particular device, that mux selection is reserved on that device and must not be used.

CAUTION

If a reserved GPIO mux configuration that is not mapped to either a peripheral or GPIO mode is selected, the state of the pin is undefined and the pin is driven. Unimplemented configurations are for future expansion and must not be selected. In the device mux table (see the data sheet), these options are indicated as Reserved or left blank.

Some peripherals can be assigned to more than one pin by way of the mux registers. For example, OUTPUTXBAR1 can be assigned to GPIOs p, q, or r (where p, q, and r are example GPIO numbers), depending on individual system requirements. An example of this is shown in Table 15-8.

Note

The following table is for example only. Bit ranges cannot correspond to OUTPUTXBAR1 on this device. See Section 15.8.1 for correct list of GPIOs and corresponding mux options for this device.

If none or more then one of the GPIO pins is configured as peripheral input pins, then that GPIO is set to a hard-wired default value.

Table 15-8. Peripheral Muxing (Multiple Pins Assigned)

GMUX Configuration	MUX Configuration	
Choice 1: GPIOp	GPyGMUX1[5:4] = 01	GPyMUX1[5:4] = 01
or Choice 2: GPIOq	GPyGMUX2[17:16] = 00	GPyMUX2[17:16] = 01
or Choice 3: GPIOr	GPyGMUX1[7:6] = 01	GPyMUX1[7:6] = 01

15.9 Internal Pullup Configuration Requirements

On reset, GPIOs are in input mode and have the internal pullups disabled. An un-driven input can float to a mid-rail voltage and cause wasted shoot-through current on the input buffer. The user must always put each GPIO in one of these configurations:

- Input mode and driven on the board by another component to a level above V_{ih} or below V_{il}
- Input mode with GPIO internal pullup enabled
- Output mode

On devices with lesser pin count packages, pull-ups on unbonded GPIOs are by default enabled to prevent floating inputs. The user must take care to avoid disabling these pullups in the application code.

On devices with larger pin count packages, the pullups for any internally unbonded GPIO must be enabled to prevent floating inputs. TI has provided functions in controlSUITE/C2000Ware that users can call to enable the pullup on any unbonded GPIO for the package in use. This function, GPIO_EnabledUnbondedIOPullups(), resides in the (Device)_Sysctrl.c file and is called by default from InitSysCtrl(). The user must take care to avoid disabling these pullups in the application code.

15.10 Software

15.10.1 GPIO Registers to Driverlib Functions

Table 15-9. GPIO Registers to Driverlib Functions

File	Driverlib Function
GPACTRL	
gpio.c	GPIO_setQualificationPeriod
GPAQSEL1	
gpio.c	GPIO_setQualificationMode
gpio.c	GPIO_getQualificationMode
GPAQSEL2	
-	See GPAQSEL1
GPAMUX1	
gpio.c	GPIO_setPinConfig
GPAMUX2	
-	See GPAMUX1
GPAPUD	
gpio.c	GPIO_setPadConfig
gpio.c	GPIO_getPadConfig
GPAINV	
gpio.c	GPIO_setPadConfig
gpio.c	GPIO_getPadConfig
GPAODR	
gpio.c	GPIO_setPadConfig
gpio.c	GPIO_getPadConfig
GPAGMUX1	
gpio.c	GPIO_setPinConfig
GPAGMUX2	
-	See GPAGMUX1
GPACSEL1	
gpio.c	GPIO_setControllerCore
GPACSEL2	
-	See GPACSEL1

Table 15-9. GPIO Registers to Driverlib Functions (continued)

File	Driverlib Function
GPACSEL3	
-	See GPACSEL1
GPACSEL4	
-	See GPACSEL1
GPALOCK	
gpio.h	GPIO_lockPortConfig
gpio.h	GPIO_unlockPortConfig
GPACR	
gpio.h	GPIO_commitPortConfig
GPBCTRL	
-	See GPACTRL
GPBQSEL1	
-	See GPAQSEL1
GPBQSEL2	
-	See GPAQSEL1
GPBMUX1	
-	See GPAMUX1
GPBMUX2	
-	See GPAMUX1
GPBPUD	
-	See GPAPUD
GPBINV	
-	See GPAINV
GPBODR	
-	See GPAODR
GPBGMUX1	
-	See GPAGMUX1
GPBGMUX2	
-	See GPAGMUX1
GPBCSEL1	
-	See GPACSEL1
GPBCSEL2	
-	See GPACSEL1
GPBCSEL3	
-	See GPACSEL1
GPBCSEL4	
-	See GPACSEL1
GPBLOCK	
-	See GPALOCK
GPBCR	
-	See GPACR
GPCCTRL	
-	See GPACTRL
GPCQSEL1	
-	See GPAQSEL1

Table 15-9. GPIO Registers to Driverlib Functions (continued)

File	Driverlib Function
GPCQSEL2	
-	See GPAQSEL1
GPCMUX1	
-	See GPAMUX1
GPCMUX2	
-	See GPAMUX1
GPCPUD	
-	See GPAPUD
GPCINV	
-	See GPAINV
GPCODR	
-	See GPAODR
GPCGMUX1	
-	See GPAGMUX1
GPCGMUX2	
-	See GPAGMUX1
GPCCSEL1	
-	See GPACSEL1
GPCCSEL2	
-	See GPACSEL1
GPCCSEL3	
-	See GPACSEL1
GPCCSEL4	
-	See GPACSEL1
GPCLOCK	
-	See GPALOCK
GPCCR	
-	See GPACR
GPDCTRL	
-	See GPACTRL
GPDQSEL1	
-	See GPAQSEL1
GPDQSEL2	
-	See GPAQSEL1
GPDMUX1	
-	See GPAMUX1
GPDMUX2	
-	See GPAMUX1
GPDPUD	
-	See GPAPUD
GPDINV	
-	See GPAINV
GPDODR	
-	See GPAODR
GPDGMUX1	

Table 15-9. GPIO Registers to Driverlib Functions (continued)

File	Driverlib Function
-	See GPAGMUX1
GPDGMUX2	
-	See GPAGMUX1
GPDCSEL1	
-	See GPACSEL1
GPDCSEL2	
-	See GPACSEL1
GPDCSEL4	
-	See GPACSEL1
GPDLOCK	
-	See GPALOCK
GPDCR	
-	See GPACR
GPFCTRL	
-	See GPACTRL
GPFQSEL1	
-	See GPAQSEL1
GPFQSEL2	
-	See GPAQSEL1
GPFMUX1	
-	See GPAMUX1
GPFMUX2	
-	See GPAMUX1
GPFPUD	
-	See GPAPUD
GPFINV	
-	See GPAINV
GPFAMSEL	
-	
GPFGMUX1	
-	See GPAGMUX1
GPFGMUX2	
-	See GPAGMUX1
GPFCSEL1	
-	See GPACSEL1
GPFCSEL2	
-	See GPACSEL1
GPFCSEL3	
-	See GPACSEL1
GPFCSEL4	
-	See GPACSEL1
GPFLOCK	
-	See GPALOCK
GPFCR	
-	See GPACR

Table 15-9. GPIO Registers to Driverlib Functions (continued)

File	Driverlib Function
GPGCTRL	
-	See GPACTRL
GPGQSEL1	
-	See GPAQSEL1
GPGQSEL2	
-	See GPAQSEL1
GPGMUX1	
-	See GPAMUX1
GPGMUX2	
-	See GPAMUX1
GPGPUD	
-	See GPAPUD
GPGINV	
-	See GPAINV
GPGODR	
-	See GPAODR
GPGAMSEL	
-	
GPGGMUX1	
-	See GPAGMUX1
GPGGMUX2	
-	See GPAGMUX1
GPGCSEL1	
-	See GPACSEL1
GPGCSEL2	
-	See GPACSEL1
GPGCSEL3	
-	See GPACSEL1
GPGCSEL4	
-	See GPACSEL1
GPGLOCK	
-	See GPALOCK
GPGCR	
-	See GPACR
GPHCTRL	
-	See GPACTRL
GPHQSEL1	
-	See GPAQSEL1
GPHQSEL2	
-	See GPAQSEL1
GPHMUX1	
-	See GPAMUX1
GPHMUX2	
-	See GPAMUX1
GPHPUD	

Table 15-9. GPIO Registers to Driverlib Functions (continued)

File	Driverlib Function
-	See GPAPUD
GPHINV	
-	See GPAINV
GPHODR	
-	See GPAODR
GPHAMSEL	
-	
GPHGMUX1	
-	See GPAGMUX1
GPHGMUX2	
-	See GPAGMUX1
GPHCSEL1	
-	See GPACSEL1
GPHCSEL2	
-	See GPACSEL1
GPHCSEL3	
-	See GPACSEL1
GPHCSEL4	
-	See GPACSEL1
GPHLOCK	
-	See GPALOCK
GPHCR	
-	See GPACR
GPADAT	
gpio.h	GPIO_readPin
gpio.h	GPIO_readPortData
gpio.h	GPIO_writePortData
GPASET	
gpio.h	GPIO_writePin
gpio.h	GPIO_setPortPins
GPACLEAR	
gpio.h	GPIO_writePin
gpio.h	GPIO_clearPortPins
GPATOGGLE	
gpio.h	GPIO_togglePin
gpio.h	GPIO_togglePortPins
GPADIR	
gpio.c	GPIO_setDirectionMode
gpio.c	GPIO_getDirectionMode
GPBDAT	
-	See GPADAT
GPBSET	
-	See GPASET
GPBCLEAR	
-	See GPACLEAR

Table 15-9. GPIO Registers to Driverlib Functions (continued)

File	Driverlib Function
GPBTOGGLE	
-	See GPATOGGLE
GPBDIR	
-	See GPADIR
GPCDAT	
-	See GPADAT
GPCSET	
-	See GPASET
GPCCLEAR	
-	See GPACLEAR
GPCTOGGLE	
-	See GPATOGGLE
GPCDIR	
-	See GPADIR
GPDDAT	
-	See GPADAT
GPDSET	
-	See GPASET
GPDCLEAR	
-	See GPACLEAR
GPDTOGGLE	
-	See GPATOGGLE
GPDDIR	
-	See GPADIR
GPFDAT	
-	See GPADAT
GPGDAT	
-	See GPADAT
GPGSET	
-	See GPASET
GPGCLEAR	
-	See GPACLEAR
GPGTOGGLE	
-	See GPATOGGLE
GPGDIR	
-	See GPADIR
GPHDAT	
-	See GPADAT
GPHSET	
-	See GPASET
GPHCLEAR	
-	See GPACLEAR
GPHTOGGLE	
-	See GPATOGGLE
GPHDIR	

Table 15-9. GPIO Registers to Driverlib Functions (continued)

File	Driverlib Function
-	See GPADIR
GPADAT_R	
-	
GPBDAT_R	
-	
GPCDAT_R	
-	
GPDDAT_R	
-	
GPFDAT_R	
-	
GPGDAT_R	
-	
GPHDAT_R	
-	

15.10.2 GPIO Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/gpio

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

15.10.2.1 Device GPIO Toggle - SINGLE_CORE

FILE: gpio_ex1_toggle.c

Configures the device GPIO through the sysconfig file. The GPIO pin is toggled in the infinite loop.

15.10.2.2 XINT/XBAR example - SINGLE_CORE

FILE: gpio_ex2_interrupt.c

This example demonstrates the XINT feature in SysConfig by using it in conjunction with the input and output XBARs. The GPIO is toggled and connected to the input XBAR, while simultaneously triggering an external interrupt. The interrupt increments a counter which can be observed in the watch window. In addition to triggering an interrupt, the input signal is routed from the input XBAR to the output XBAR so that the output can be observed via oscilloscope or logic analyzer on a separate pin.

Watch Variables

- *counter* - Number of interrupts generated

15.10.3 LED Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/led

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

15.10.3.1 LED Blinky Example - MULTI_CORE

FILE: led_ex1_blinky_cpu1_cpu3_multi_c29x1.c

This example demonstrates how to blink a LED using CPU1 and blink another LED using CPU3 (led_ex1_blinky_cpu1_cpu3_multi_c29x3.c).

When using CCS for debugging this Multi-core example, after launching the debug session,

- Connect to CPU1 and load only the c29x1.out.
- After the program is loaded, run CPU1.
- Once c29x1 configures and releases CPU3 out of reset, the program stops.
- Connect to CPU3 target now. c29x3.out would have started execution as soon as it is released from reset.
- In case of RAM configuration, restart the CPU3 target and load the symbols.

For FLASH configuration, this example is run in FLASH BANKMODE2, where CPU3 has access to FLASH (FRI-2). Refer to the Flash Plugin documentation to know about changing FLASH BANKMODEs and more. Additionally, the CPAx and CDAx RAMs are used for allocating various RAM sections. *External Connections*

- None.

Watch Variables

- None.

15.10.3.2 LED Blinky Example (CPU1,CPU3) - MULTI_CORE

FILE: led_ex1_blinky_cpu1_cpu3_multi_c29x3.c

This example demonstrates how to blink a LED using CPU1 and blink another LED using CPU3 (led_ex1_blinky_cpu1_cpu3_multi_c29x3.c).

When using CCS for debugging this Multi-core example, after launching the debug session,

- Connect to CPU1 and first load the c29x3.out. (When loaded to FLASH, this might not halt at main due to an error, since CPU1 cannot execute from CPU3 FLASH memory (FRI-2)).
- c29x1.out can be loaded next.
- After the program is loaded, run CPU1.
- Once c29x1 configures and releases CPU3 out of reset, the program stops.
- Connect to CPU3 target now. c29x3.out would have started execution as soon as it is released from reset.
- In case of RAM configuration, restart the CPU3 target and load the symbols.
- In case of FLASH, do a Power-on-Reset, connect to CPU1/ CPU3 and load the respective symbols.

For FLASH configuration, this example is run in FLASH BANKMODE2, where CPU3 has access to FLASH (FRI-2). Refer to the Flash Plugin documentation to know about changing FLASH BANKMODEs and more. Additionally, the CPAx and CDAx RAMs are used for allocating various RAM sections. *External Connections*

- None.

Watch Variables

- None.

15.10.3.3 LED Blinky example - SINGLE_CORE

FILE: led_ex1_blinky.c

This example demonstrates how to blink an LED. The device GPIO is configured through the sysconfig file. The GPIO pin is toggled in an infinite loop.

15.10.3.4 LED Blinky Example (CPU1|CPU2|CPU3) - MULTI_CORE

FILE: led_ex2_blinky_cpu1_cpu2_cpu3_multi_c29x1.c

This example demonstrates how to configure LEDs using CPU1 and blink two LEDs using CPU2 (led_ex2_blinky_cpu1_cpu2_cpu3_multi_c29x2.c) and CPU3 (led_ex2_blinky_cpu1_cpu2_cpu3_multi_c29x3.c).

When using CCS for debugging this Multi-core example, after launching the debug session,

- Connect to CPU1 and load only the c29x1.out.
- After the program is loaded, run CPU1.
- Once c29x1 disables lockstep, configures and releases CPU2 & CPU3 out of reset, the program stops.
- Connect to the CPU2 target now. c29x2.out would have started execution as soon as it is released from reset.

- Connect to the CPU3 target now. c29x3.out would have started execution as soon as it is released from reset.
- In case of RAM configuration, restart CPU2, CPU3 targets and load the symbols.

Note: Since CPU2 does not have FLASH access, the CPU2 application code is loaded to CPU1's FLASH, which is copied to CPU2 RAM by the CPU1 application. The CPU3 application code is loaded to its Flash (FRI-2) in FLASH BANKMODE2.

External Connections

- None.

Watch Variables

- None.

15.10.3.5 LED Blinky Example (CPU2) - MULTI_CORE

FILE: led_ex2_blinky_cpu1_cpu2_cpu3_multi_c29x2.c

This example demonstrates how to configure LEDs using CPU1 and blink two LEDs using CPU2 (led_ex2_blinky_cpu1_cpu2_cpu3_multi_c29x2.c) and CPU3 (led_ex2_blinky_cpu1_cpu2_cpu3_multi_c29x3.c).

When using CCS for debugging this Multi-core example, after launching the debug session,

- Connect to CPU1 and first load the c29x3.out, c29x2.out.
- When loaded to FLASH, these might not halt at main due to an error.
- c29x1.out can be loaded next.
- After the program is loaded, run CPU1.
- Once c29x1 disables lockstep, configures and releases CPU2 & CPU3 out of reset, the program stops.
- Connect to the CPU2 target now. c29x2.out would have started execution as soon as it is released from reset.
- Connect to the CPU3 target now. c29x3.out would have started execution as soon as it is released from reset.
- In case of RAM configuration, restart CPU2, CPU3 targets and load the symbols.
- In case of FLASH, do a Power-on-Reset, connect to CPU1, CPU2, CPU3 and load the respective symbols.

Note: Since CPU2 does not have FLASH access, the CPU2 application code is loaded to CPU1's FLASH, which is copied to CPU2 RAM by the CPU1 application. The CPU3 application code is loaded to its Flash (FRI-2) in FLASH BANKMODE2.

External Connections

- None.

Watch Variables

- None.

15.10.3.6 LED Blinky Example (CPU3) - MULTI_CORE

FILE: led_ex2_blinky_cpu1_cpu2_cpu3_multi_c29x3.c

This example demonstrates how to configure LEDs using CPU1 and blink two LEDs using CPU2 (led_ex2_blinky_cpu1_cpu2_cpu3_multi_c29x2.c) and CPU3 (led_ex2_blinky_cpu1_cpu2_cpu3_multi_c29x3.c).

When using CCS for debugging this Multi-core example, after launching the debug session,

- Connect to CPU1 and first load the c29x3.out, c29x2.out.
- When loaded to FLASH, these might not halt at main due to an error.
- c29x1.out can be loaded next.
- After the program is loaded, run CPU1.
- Once c29x1 disables lockstep, configures and releases CPU2 & CPU3 out of reset, the program stops.
- Connect to the CPU2 target now. c29x2.out would have started execution as soon as it is released from reset.

- Connect to the CPU3 target now. c29x3.out would have started execution as soon as it is released from reset.
- In case of RAM configuration, restart CPU2, CPU3 targets and load the symbols.
- In case of FLASH, do a Power-on-Reset, connect to CPU1, CPU2, CPU3 and load the respective symbols.

Note: Since CPU2 does not have FLASH access, the CPU2 application code is loaded to CPU1's FLASH, which is copied to CPU2 RAM by the CPU1 application. The CPU3 application code is loaded to its Flash (FRI-2) in FLASH BANKMODE2.

External Connections

- None.

Watch Variables

- None.

15.11 GPIO Registers

This section describes the GPIO Registers.

15.11.1 GPIO Base Address Table

Table 15-10. GPIO Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
GPIO_CTRL_REGS [1]	GPIOCTRL_BASE	0x3019_0000	-	YES	YES	YES	-	-	YES	YES
GPIO_DATA_REGS	GPIODATA_BASE	0x3026_8000	-	YES	YES	YES	-	-	-	YES
GPIO_DATA_READ_REGS	GPIODATAREAD_BASE	0x3026_9000	-	YES	YES	YES	-	-	-	YES

(1) Registers writeable by CPU1.LINK0, CPU1.LINK1, CPU1.LINK2 only. All CPUs can read all registers in all LINKs. Debug write access only allowed if Zone0 or Zone1 are enabled for full debug by all CPUs. Debug reads always allowed. Register Read/Write access by HSM.

15.11.2 GPIO_CTRL_REGS Registers

Table 15-11 lists the memory-mapped registers for the GPIO_CTRL_REGS registers. All register offset addresses not listed in Table 15-11 should be considered as reserved locations and the register contents should not be modified.

Table 15-11. GPIO_CTRL_REGS Registers

Offset	Acronym	Register Name	Protection
0h	GPACTRL	GPIO A Qualification Sampling Period Control (GPIO0 to 31)	
4h	GPAQSEL1	GPIO A Qualifier Select 1 Register (GPIO0 to 15)	
8h	GPAQSEL2	GPIO A Qualifier Select 2 Register (GPIO16 to 31)	
Ch	GPAMUX1	GPIO A Mux 1 Register (GPIO0 to 15)	
10h	GPAMUX2	GPIO A Mux 2 Register (GPIO16 to 31)	
18h	GPAPUD	GPIO A Pull Up Disable Register (GPIO0 to 31)	
20h	GPAINV	GPIO A Input Polarity Invert Registers (GPIO0 to 31)	
24h	GPAODR	GPIO A Open Drain Output Register (GPIO0 to GPIO31)	
40h	GPAGMUX1	GPIO A Peripheral Group Mux (GPIO0 to 15)	
44h	GPAGMUX2	GPIO A Peripheral Group Mux (GPIO16 to 31)	
50h	GPACSEL1	GPIO A Core Select Register (GPIO0 to 7)	
54h	GPACSEL2	GPIO A Core Select Register (GPIO8 to 15)	
58h	GPACSEL3	GPIO A Core Select Register (GPIO16 to 23)	
5Ch	GPACSEL4	GPIO A Core Select Register (GPIO24 to 31)	
78h	GPALOCK	GPIO A Lock Configuration Register (GPIO0 to 31)	
7Ch	GPACR	GPIO A Lock Commit Register (GPIO0 to 31)	
80h	GPBCTRL	GPIO B Qualification Sampling Period Control (GPIO32 to 63)	
84h	GPBQSEL1	GPIO B Qualifier Select 1 Register (GPIO32 to 47)	
88h	GPBQSEL2	GPIO B Qualifier Select 2 Register (GPIO48 to 63)	
8Ch	GPBMUX1	GPIO B Mux 1 Register (GPIO32 to 47)	
90h	GPBMUX2	GPIO B Mux 2 Register (GPIO48 to 63)	
98h	GPBPUD	GPIO B Pull Up Disable Register (GPIO32 to 63)	
A0h	GPBINV	GPIO B Input Polarity Invert Registers (GPIO32 to 63)	
A4h	GPBODR	GPIO B Open Drain Output Register (GPIO32 to GPIO63)	
C0h	GPBGMUX1	GPIO B Peripheral Group Mux (GPIO32 to 47)	
C4h	GPBGMUX2	GPIO B Peripheral Group Mux (GPIO48 to 63)	
D0h	GPBCSEL1	GPIO B Core Select Register (GPIO32 to 39)	
D4h	GPBCSEL2	GPIO B Core Select Register (GPIO40 to 47)	
D8h	GPBCSEL3	GPIO B Core Select Register (GPIO48 to 55)	
DCh	GPBCSEL4	GPIO B Core Select Register (GPIO56 to 63)	
F8h	GPBLOCK	GPIO B Lock Configuration Register (GPIO32 to 63)	
FCh	GPBCR	GPIO B Lock Commit Register (GPIO32 to 63)	
100h	GPCCTRL	GPIO C Qualification Sampling Period Control (GPIO64 to 95)	
104h	GPCQSEL1	GPIO C Qualifier Select 1 Register (GPIO64 to 79)	
108h	GPCQSEL2	GPIO C Qualifier Select 2 Register (GPIO80 to 95)	
10Ch	GPCMUX1	GPIO C Mux 1 Register (GPIO64 to 79)	
110h	GPCMUX2	GPIO C Mux 2 Register (GPIO80 to 95)	
118h	GPCPUD	GPIO C Pull Up Disable Register (GPIO64 to 95)	

Table 15-11. GPIO_CTRL_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
120h	GPCINV	GPIO C Input Polarity Invert Registers (GPIO64 to 95)	
124h	GPCODR	GPIO C Open Drain Output Register (GPIO64 to GPIO95)	
140h	GPCGMUX1	GPIO C Peripheral Group Mux (GPIO64 to 79)	
144h	GPCGMUX2	GPIO C Peripheral Group Mux (GPIO80 to 95)	
150h	GPCCSEL1	GPIO C Core Select Register (GPIO64 to 71)	
154h	GPCCSEL2	GPIO C Core Select Register (GPIO72 to 79)	
158h	GPCCSEL3	GPIO C Core Select Register (GPIO80 to 87)	
15Ch	GPCCSEL4	GPIO C Core Select Register (GPIO88 to 95)	
178h	GPCLOCK	GPIO C Lock Configuration Register (GPIO64 to 95)	
17Ch	GPCCR	GPIO C Lock Commit Register (GPIO64 to 95)	
180h	GPDCTRL	GPIO D Qualification Sampling Period Control (GPIO96 to 127)	
184h	GPDQSEL1	GPIO D Qualifier Select 1 Register (GPIO96 to 111)	
188h	GPDQSEL2	GPIO D Qualifier Select 2 Register (GPIO112 to 127)	
18Ch	GPDMUX1	GPIO D Mux 1 Register (GPIO96 to 111)	
190h	GPDMUX2	GPIO D Mux 2 Register (GPIO112 to 127)	
198h	GPDPUD	GPIO D Pull Up Disable Register (GPIO96 to 127)	
1A0h	GPDINV	GPIO D Input Polarity Invert Registers (GPIO96 to 127)	
1A4h	GPDODR	GPIO D Open Drain Output Register (GPIO96 to GPIO127)	
1C0h	GPDGMUX1	GPIO D Peripheral Group Mux (GPIO96 to 111)	
1C4h	GPDGMUX2	GPIO D Peripheral Group Mux (GPIO112 to 127)	
1D0h	GPDCSEL1	GPIO D Core Select Register (GPIO96 to 103)	
1D4h	GPDCSEL2	GPIO D Core Select Register (GPIO104 to 111)	
1D8h	GPDCSEL3	GPIO D Core Select Register (GPIO112 to 119)	
1DCh	GPDCSEL4	GPIO D Core Select Register (GPIO120 to 127)	
1F8h	GPDLOCK	GPIO D Lock Configuration Register (GPIO96 to 127)	
1FCh	GPDCR	GPIO D Lock Commit Register (GPIO96 to 127)	
280h	GPFCTRL	GPIO F Qualification Sampling Period Control (GPIO160 to 191)	
284h	GPFQSEL1	GPIO F Qualifier Select 1 Register (GPIO160 to 168)	
288h	GPFQSEL2	GPIO F Qualifier Select 2 Register (GPIO176 to 191)	
28Ch	GPFMUX1	GPIO F Mux 1 Register (GPIO160 to 175)	
290h	GPFMUX2	GPIO F Mux 2 Register (GPIO176 to 191)	
298h	GPFPUD	GPIO F Pull Up Disable Register (GPIO160 to 191)	
2A0h	GPFINV	GPIO F Input Polarity Invert Registers (GPIO160 to 191)	
2A8h	GPFAMSEL	GPIO F Analog Mode Select register (GPIO160 to GPIO191)	
2C0h	GPFGMUX1	GPIO F Peripheral Group Mux (GPIO160 to 175)	
2C4h	GPFGMUX2	GPIO F Peripheral Group Mux (GPIO176 to 191)	
2D0h	GPFCSEL1	GPIO F Core Select Register (GPIO160 to 167)	
2D4h	GPFCSEL2	GPIO F Core Select Register (GPIO168 to 175)	
2D8h	GPFCSEL3	GPIO F Core Select Register (GPIO176 to 183)	
2DCh	GPFCSEL4	GPIO F Core Select Register (GPIO184 to 191)	
2F8h	GPFLOCK	GPIO F Lock Configuration Register (GPIO160 to 191)	
2FCh	GPFCR	GPIO F Lock Commit Register (GPIO160 to 191)	

Table 15-11. GPIO_CTRL_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
300h	GPGCTRL	GPIO G Qualification Sampling Period Control (GPIO192 to 223)	
304h	GPGQSEL1	GPIO G Qualifier Select 1 Register (GPIO192 to 207)	
308h	GPGQSEL2	GPIO G Qualifier Select 2 Register (GPIO208 to 223)	
30Ch	GPGMUX1	GPIO G Mux 1 Register (GPIO192 to 207)	
310h	GPGMUX2	GPIO G Mux 2 Register (GPIO208 to 223)	
318h	GPGPUD	GPIO G Pull Up Disable Register (GPIO192 to 223)	
320h	GPGINV	GPIO G Input Polarity Invert Registers (GPIO192 to 223)	
324h	GPGODR	GPIO G Open Drain Output Register (GPIO192 to 223)	
328h	GPGAMSEL	GPIO G Analog Mode Select register (GPIO192 to 223)	
340h	GPGGMUX1	GPIO G Peripheral Group Mux (GPIO192 to 207)	
344h	GPGGMUX2	GPIO G Peripheral Group Mux (GPIO208 to 223)	
350h	GPGCSEL1	GPIO G Core Select Register (GPIO192 to 199)	
354h	GPGCSEL2	GPIO G Core Select Register (GPIO200 to 207)	
358h	GPGCSEL3	GPIO G Core Select Register (GPIO208 to 215)	
35Ch	GPGCSEL4	GPIO G Core Select Register (GPIO216 to 223)	
378h	GPGLOCK	GPIO G Lock Configuration Register (GPIO192 to 223)	
37Ch	GPGCR	GPIO G Lock Commit Register (GPIO192 to 223)	
380h	GPHCTRL	GPIO H Qualification Sampling Period Control (GPIO224 to 255)	
384h	GPHQSEL1	GPIO H Qualifier Select 1 Register (GPIO224 to 239)	
388h	GPHQSEL2	GPIO H Qualifier Select 2 Register (GPIO240 to 255)	
38Ch	GPHMUX1	GPIO H Mux 1 Register (GPIO224 to 239)	
390h	GPHMUX2	GPIO H Mux 2 Register (GPIO240 to 255)	
398h	GPHPUD	GPIO H Pull Up Disable Register (GPIO224 to 255)	
3A0h	GPHINV	GPIO H Input Polarity Invert Registers (GPIO224 to 255)	
3A4h	GPHODR	GPIO H Open Drain Output Register (GPIO224 to GPIO255)	
3A8h	GPHAMSEL	GPIO H Analog Mode Select register (GPIO224 to GPIO255)	
3C0h	GPHGMUX1	GPIO H Peripheral Group Mux (GPIO224 to 239)	
3C4h	GPHGMUX2	GPIO H Peripheral Group Mux (GPIO240 to 255)	
3D0h	GPHCSEL1	GPIO H Core Select Register (GPIO224 to 231)	
3D4h	GPHCSEL2	GPIO H Core Select Register (GPIO232 to 239)	
3D8h	GPHCSEL3	GPIO H Core Select Register (GPIO240 to 247)	
3DCh	GPHCSEL4	GPIO H Core Select Register (GPIO248 to 255)	
3F8h	GPHLOCK	GPIO H Lock Configuration Register (GPIO224 to 255)	
3FCh	GPHCR	GPIO H Lock Commit Register (GPIO224 to 255)	

Complex bit access types are encoded to fit into small table cells. Table 15-12 shows the codes that are used for access types in this section.

Table 15-12. GPIO_CTRL_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		

Table 15-12. GPIO_CTRL_REGS Access Type Codes (continued)

Access Type	Code	Description
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

15.11.2.1 GPACTRL Register (Offset = 0h) [Reset = 00000000h]

GPACTRL is shown in Figure 15-4 and described in Table 15-13.

Return to the Summary Table.

GPIO A Qualification Sampling Period Control (GPIO0 to 31)

Figure 15-4. GPACTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUALPRD3								QUALPRD2								QUALPRD1								QUALPRD0							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 15-13. GPACTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	QUALPRD3	R/W	0h	Qualification sampling period for GPIO24 to GPIO31: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
23-16	QUALPRD2	R/W	0h	Qualification sampling period for GPIO16 to GPIO23: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
15-8	QUALPRD1	R/W	0h	Qualification sampling period for GPIO8 to GPIO15: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
7-0	QUALPRD0	R/W	0h	Qualification sampling period for GPIO0 to GPIO7: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn

15.11.2.2 GPAQSEL1 Register (Offset = 4h) [Reset = 00000000h]

GPAQSEL1 is shown in Figure 15-5 and described in Table 15-14.

Return to the Summary Table.

GPIO A Qualifier Select 1 Register (GPIO0 to 15)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 15-5. GPAQSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO15		GPIO14		GPIO13		GPIO12		GPIO11		GPIO10		GPIO9		GPIO8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO7		GPIO6		GPIO5		GPIO4		GPIO3		GPIO2		GPIO1		GPIO0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-14. GPAQSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO15	R/W	0h	Select input qualification type for GPIO15: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
29-28	GPIO14	R/W	0h	Select input qualification type for GPIO14: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
27-26	GPIO13	R/W	0h	Select input qualification type for GPIO13: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
25-24	GPIO12	R/W	0h	Select input qualification type for GPIO12: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
23-22	GPIO11	R/W	0h	Select input qualification type for GPIO11: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
21-20	GPIO10	R/W	0h	Select input qualification type for GPIO10: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-14. GPAQSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	GPIO9	R/W	0h	Select input qualification type for GPIO9: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
17-16	GPIO8	R/W	0h	Select input qualification type for GPIO8: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
15-14	GPIO7	R/W	0h	Select input qualification type for GPIO7: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
13-12	GPIO6	R/W	0h	Select input qualification type for GPIO6: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
11-10	GPIO5	R/W	0h	Select input qualification type for GPIO5: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
9-8	GPIO4	R/W	0h	Select input qualification type for GPIO4: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
7-6	GPIO3	R/W	0h	Select input qualification type for GPIO3: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
5-4	GPIO2	R/W	0h	Select input qualification type for GPIO2: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	GPIO1	R/W	0h	Select input qualification type for GPIO1: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-14. GPAQSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO0	R/W	0h	Select input qualification type for GPIO0: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

15.11.2.3 GPAQSEL2 Register (Offset = 8h) [Reset = 00000000h]

GPAQSEL2 is shown in Figure 15-6 and described in Table 15-15.

Return to the Summary Table.

GPIO A Qualifier Select 2 Register (GPIO16 to 31)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 15-6. GPAQSEL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO31		GPIO30		GPIO29		GPIO28		GPIO27		GPIO26		GPIO25		GPIO24	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO23		GPIO22		GPIO21		GPIO20		GPIO19		GPIO18		GPIO17		GPIO16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-15. GPAQSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO31	R/W	0h	Select input qualification type for GPIO31: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
29-28	GPIO30	R/W	0h	Select input qualification type for GPIO30: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
27-26	GPIO29	R/W	0h	Select input qualification type for GPIO29: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
25-24	GPIO28	R/W	0h	Select input qualification type for GPIO28: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
23-22	GPIO27	R/W	0h	Select input qualification type for GPIO27: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
21-20	GPIO26	R/W	0h	Select input qualification type for GPIO26: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-15. GPAQSEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	GPIO25	R/W	0h	Select input qualification type for GPIO25: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
17-16	GPIO24	R/W	0h	Select input qualification type for GPIO24: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
15-14	GPIO23	R/W	0h	Select input qualification type for GPIO23: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
13-12	GPIO22	R/W	0h	Select input qualification type for GPIO22: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
11-10	GPIO21	R/W	0h	Select input qualification type for GPIO21: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
9-8	GPIO20	R/W	0h	Select input qualification type for GPIO20: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
7-6	GPIO19	R/W	0h	Select input qualification type for GPIO19: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
5-4	GPIO18	R/W	0h	Select input qualification type for GPIO18: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	GPIO17	R/W	0h	Select input qualification type for GPIO17: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-15. GPAQSEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO16	R/W	0h	Select input qualification type for GPIO16: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

15.11.2.4 GPAMUX1 Register (Offset = Ch) [Reset = 00000000h]

GPAMUX1 is shown in Figure 15-7 and described in Table 15-16.

Return to the Summary Table.

GPIO A Mux 1 Register (GPIO0 to 15)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 15-7. GPAMUX1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO15		GPIO14		GPIO13		GPIO12		GPIO11		GPIO10		GPIO9		GPIO8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO7		GPIO6		GPIO5		GPIO4		GPIO3		GPIO2		GPIO1		GPIO0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-16. GPAMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO15	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO14	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO13	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO12	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO11	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO10	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO9	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO8	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO7	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO6	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO5	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO4	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO3	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO2	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO1	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-16. GPAMUX1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO0	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.5 GPAMUX2 Register (Offset = 10h) [Reset = 00000000h]

GPAMUX2 is shown in Figure 15-8 and described in Table 15-17.

Return to the Summary Table.

GPIO A Mux 2 Register (GPIO16 to 31)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 15-8. GPAMUX2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO31		GPIO30		GPIO29		GPIO28		GPIO27		GPIO26		GPIO25		GPIO24	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO23		GPIO22		GPIO21		GPIO20		GPIO19		GPIO18		GPIO17		GPIO16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-17. GPAMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO31	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO30	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO29	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO28	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO27	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO26	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO25	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO24	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO23	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO22	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO21	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO20	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO19	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO18	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO17	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-17. GPAMUX2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO16	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.6 GPAPUD Register (Offset = 18h) [Reset = FFFFFFFFh]

GPAPUD is shown in Figure 15-9 and described in Table 15-18.

Return to the Summary Table.

GPIO A Pull Up Disable Register (GPIO0 to 31)
Disables the Pull-Up on GPIO.
0: Enables the Pull-Up.
1: Disables the Pull-Up.
Reading the register returns the current value of the register setting.
Note:
[1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low.

Figure 15-9. GPAPUD Register

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 15-18. GPAPUD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO31	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
30	GPIO30	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
29	GPIO29	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
28	GPIO28	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
27	GPIO27	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
26	GPIO26	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
25	GPIO25	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
24	GPIO24	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
23	GPIO23	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
22	GPIO22	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
21	GPIO21	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn

Table 15-18. GPAPUD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO20	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
19	GPIO19	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
18	GPIO18	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
17	GPIO17	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
16	GPIO16	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
15	GPIO15	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
14	GPIO14	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
13	GPIO13	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
12	GPIO12	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
11	GPIO11	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
10	GPIO10	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
9	GPIO9	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
8	GPIO8	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
7	GPIO7	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
6	GPIO6	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
5	GPIO5	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
4	GPIO4	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
3	GPIO3	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
2	GPIO2	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
1	GPIO1	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
0	GPIO0	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn

15.11.2.7 GPAINV Register (Offset = 20h) [Reset = 00000000h]

GPAINV is shown in Figure 15-10 and described in Table 15-19.

Return to the Summary Table.

GPIO A Input Polarity Invert Registers (GPIO0 to 31)
Selects between non-inverted and inverted GPIO input to the device.
0: selects non-inverted GPIO input
1: selects inverted GPIO input
Reading the register returns the current value of the register setting.

Figure 15-10. GPAINV Register

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-19. GPAINV Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO31	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
30	GPIO30	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
29	GPIO29	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
28	GPIO28	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
27	GPIO27	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
26	GPIO26	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
25	GPIO25	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
24	GPIO24	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
23	GPIO23	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
22	GPIO22	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
21	GPIO21	R/W	0h	Input inversion control for this pin Reset type: SYSRSn

Table 15-19. GPAINV Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO20	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
19	GPIO19	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
18	GPIO18	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
17	GPIO17	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
16	GPIO16	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
15	GPIO15	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
14	GPIO14	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
13	GPIO13	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
12	GPIO12	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
11	GPIO11	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
10	GPIO10	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
9	GPIO9	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
8	GPIO8	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
7	GPIO7	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
6	GPIO6	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
5	GPIO5	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
4	GPIO4	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
3	GPIO3	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
2	GPIO2	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
1	GPIO1	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
0	GPIO0	R/W	0h	Input inversion control for this pin Reset type: SYSRSn

15.11.2.8 GPAODR Register (Offset = 24h) [Reset = 00000000h]

GPAODR is shown in Figure 15-11 and described in Table 15-20.

Return to the Summary Table.

GPIO A Open Drain Output Register (GPIO0 to GPIO31)
Selects between normal and open-drain output for the GPIO pin.
0: Normal Output
1: Open Drain Output
Reading the register returns the current value of the register setting.
Note:
[1] In the Open Drain output mode, if the buffer is configured for output mode, a 0 value to be driven out comes out on the on the PAD while a 1 value to be driven out tri-states the buffer.

Figure 15-11. GPAODR Register

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-20. GPAODR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO31	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
30	GPIO30	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
29	GPIO29	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
28	GPIO28	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
27	GPIO27	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
26	GPIO26	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
25	GPIO25	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
24	GPIO24	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
23	GPIO23	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
22	GPIO22	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn

Table 15-20. GPAODR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	GPIO21	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
20	GPIO20	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
19	GPIO19	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
18	GPIO18	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
17	GPIO17	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
16	GPIO16	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
15	GPIO15	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
14	GPIO14	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
13	GPIO13	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
12	GPIO12	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
11	GPIO11	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
10	GPIO10	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
9	GPIO9	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
8	GPIO8	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
7	GPIO7	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
6	GPIO6	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
5	GPIO5	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
4	GPIO4	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
3	GPIO3	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
2	GPIO2	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
1	GPIO1	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
0	GPIO0	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn

15.11.2.9 GPAGMUX1 Register (Offset = 40h) [Reset = 00000000h]

GPAGMUX1 is shown in Figure 15-12 and described in Table 15-21.

Return to the Summary Table.

GPIO A Peripheral Group Mux (GPIO0 to 15)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 15-12. GPAGMUX1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO15		GPIO14		GPIO13		GPIO12		GPIO11		GPIO10		GPIO9		GPIO8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO7		GPIO6		GPIO5		GPIO4		GPIO3		GPIO2		GPIO1		GPIO0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-21. GPAGMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO15	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO14	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO13	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO12	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO11	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO10	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO9	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO8	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO7	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO6	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO5	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO4	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO3	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO2	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO1	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-21. GPAGMUX1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO0	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.10 GPAGMUX2 Register (Offset = 44h) [Reset = 00000000h]

GPAGMUX2 is shown in Figure 15-13 and described in Table 15-22.

Return to the Summary Table.

GPIO A Peripheral Group Mux (GPIO16 to 31)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 15-13. GPAGMUX2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO31		GPIO30		GPIO29		GPIO28		GPIO27		GPIO26		GPIO25		GPIO24	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO23		GPIO22		GPIO21		GPIO20		GPIO19		GPIO18		GPIO17		GPIO16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-22. GPAGMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO31	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO30	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO29	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO28	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO27	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO26	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO25	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO24	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO23	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO22	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO21	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO20	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO19	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO18	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO17	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-22. GPAGMUX2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO16	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.11 GPACSEL1 Register (Offset = 50h) [Reset = 00000000h]

GPACSEL1 is shown in Figure 15-14 and described in Table 15-23.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-14. GPACSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO7				GPIO6				GPIO5				GPIO4			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO3				GPIO2				GPIO1				GPIO0			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-23. GPACSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO7	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO6	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO5	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO4	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO3	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO2	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO1	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO0	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.12 GPACSEL2 Register (Offset = 54h) [Reset = 00000000h]

GPACSEL2 is shown in Figure 15-15 and described in Table 15-24.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-15. GPACSEL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO15				GPIO14				GPIO13				GPIO12			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO11				GPIO10				GPIO9				GPIO8			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-24. GPACSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO15	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO14	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO13	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO12	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO11	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO10	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO9	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO8	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.13 GPACSEL3 Register (Offset = 58h) [Reset = 00000000h]

GPACSEL3 is shown in Figure 15-16 and described in Table 15-25.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-16. GPACSEL3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO23				GPIO22				GPIO21				GPIO20			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO19				GPIO18				GPIO17				GPIO16			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-25. GPACSEL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO23	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO22	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO21	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO20	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO19	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO18	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO17	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO16	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.14 GPACSEL4 Register (Offset = 5Ch) [Reset = 00000000h]

GPACSEL4 is shown in Figure 15-17 and described in Table 15-26.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-17. GPACSEL4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO31				GPIO30				GPIO29				GPIO28			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO27				GPIO26				GPIO25				GPIO24			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-26. GPACSEL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO31	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO30	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO29	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO28	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO27	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO26	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO25	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO24	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.15 GPALOCK Register (Offset = 78h) [Reset = 00000000h]

GPALOCK is shown in Figure 15-18 and described in Table 15-27.

Return to the Summary Table.

GPIO A Lock Configuration Register (GPIO0 to 31)
GPIO Configuration Lock for GPIO.
0: Bits in GPyMUX1, GPyMUX2, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed
1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin

Figure 15-18. GPALOCK Register

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-27. GPALOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO31	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
30	GPIO30	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
29	GPIO29	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
28	GPIO28	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
27	GPIO27	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
26	GPIO26	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
25	GPIO25	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
24	GPIO24	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
23	GPIO23	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
22	GPIO22	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
21	GPIO21	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn

Table 15-27. GPALOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO20	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
19	GPIO19	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
18	GPIO18	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
17	GPIO17	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
16	GPIO16	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
15	GPIO15	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
14	GPIO14	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
13	GPIO13	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
12	GPIO12	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
11	GPIO11	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
10	GPIO10	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
9	GPIO9	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
8	GPIO8	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
7	GPIO7	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
6	GPIO6	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
5	GPIO5	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
4	GPIO4	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
3	GPIO3	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
2	GPIO2	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
1	GPIO1	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
0	GPIO0	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn

15.11.2.16 GPACR Register (Offset = 7Ch) [Reset = 00000000h]

GPACR is shown in Figure 15-19 and described in Table 15-28.

Return to the Summary Table.

GPIO A Lock Commit Register (GPIO0 to 31)
GPIO Configuration Lock Commit for GPIO:
1: Locks changes to the bit in GPyLOCK register which controls the same pin
0: Bit in the GPyLOCK register which controls the same pin can be changed

Figure 15-19. GPACR Register

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 15-28. GPACR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO31	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
30	GPIO30	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
29	GPIO29	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
28	GPIO28	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
27	GPIO27	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
26	GPIO26	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
25	GPIO25	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
24	GPIO24	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
23	GPIO23	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
22	GPIO22	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
21	GPIO21	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
20	GPIO20	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn

Table 15-28. GPACR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO19	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
18	GPIO18	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
17	GPIO17	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
16	GPIO16	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
15	GPIO15	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
14	GPIO14	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
13	GPIO13	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
12	GPIO12	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
11	GPIO11	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
10	GPIO10	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
9	GPIO9	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
8	GPIO8	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
7	GPIO7	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
6	GPIO6	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
5	GPIO5	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
4	GPIO4	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
3	GPIO3	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
2	GPIO2	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
1	GPIO1	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
0	GPIO0	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn

15.11.2.17 GPBCTRL Register (Offset = 80h) [Reset = 00000000h]

GPBCTRL is shown in Figure 15-20 and described in Table 15-29.

Return to the Summary Table.

GPIO B Qualification Sampling Period Control (GPIO32 to 63)

Figure 15-20. GPBCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUALPRD3								QUALPRD2								QUALPRD1								QUALPRD0							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 15-29. GPBCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	QUALPRD3	R/W	0h	Qualification sampling period for GPIO56 to GPIO63: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
23-16	QUALPRD2	R/W	0h	Qualification sampling period for GPIO48 to GPIO55: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
15-8	QUALPRD1	R/W	0h	Qualification sampling period for GPIO40 to GPIO47: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
7-0	QUALPRD0	R/W	0h	Qualification sampling period for GPIO32 to GPIO39: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn

15.11.2.18 GPBQSEL1 Register (Offset = 84h) [Reset = 00000000h]

GPBQSEL1 is shown in Figure 15-21 and described in Table 15-30.

Return to the Summary Table.

GPIO B Qualifier Select 1 Register (GPIO32 to 47)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 15-21. GPBQSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO47		GPIO46		GPIO45		GPIO44		GPIO43		GPIO42		GPIO41		GPIO40	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO39		GPIO38		GPIO37		GPIO36		GPIO35		GPIO34		GPIO33		GPIO32	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-30. GPBQSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO47	R/W	0h	Select input qualification type for GPIO47: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
29-28	GPIO46	R/W	0h	Select input qualification type for GPIO46: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
27-26	GPIO45	R/W	0h	Select input qualification type for GPIO45: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
25-24	GPIO44	R/W	0h	Select input qualification type for GPIO44: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
23-22	GPIO43	R/W	0h	Select input qualification type for GPIO43: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
21-20	GPIO42	R/W	0h	Select input qualification type for GPIO42: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-30. GPBQSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	GPIO41	R/W	0h	Select input qualification type for GPIO41: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
17-16	GPIO40	R/W	0h	Select input qualification type for GPIO40: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
15-14	GPIO39	R/W	0h	Select input qualification type for GPIO39: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
13-12	GPIO38	R/W	0h	Select input qualification type for GPIO38: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
11-10	GPIO37	R/W	0h	Select input qualification type for GPIO37: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
9-8	GPIO36	R/W	0h	Select input qualification type for GPIO36: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
7-6	GPIO35	R/W	0h	Select input qualification type for GPIO35: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
5-4	GPIO34	R/W	0h	Select input qualification type for GPIO34: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	GPIO33	R/W	0h	Select input qualification type for GPIO33: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-30. GPBQSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO32	R/W	0h	Select input qualification type for GPIO32: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

15.11.2.19 GPBQSEL2 Register (Offset = 88h) [Reset = 00000000h]

GPBQSEL2 is shown in Figure 15-22 and described in Table 15-31.

Return to the Summary Table.

GPIO B Qualifier Select 2 Register (GPIO48 to 63)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 15-22. GPBQSEL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO63		GPIO62		GPIO61		GPIO60		GPIO59		GPIO58		GPIO57		GPIO56	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO55		GPIO54		GPIO53		GPIO52		GPIO51		GPIO50		GPIO49		GPIO48	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-31. GPBQSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO63	R/W	0h	Select input qualification type for GPIO63: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
29-28	GPIO62	R/W	0h	Select input qualification type for GPIO62: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
27-26	GPIO61	R/W	0h	Select input qualification type for GPIO61: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
25-24	GPIO60	R/W	0h	Select input qualification type for GPIO60: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
23-22	GPIO59	R/W	0h	Select input qualification type for GPIO59: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
21-20	GPIO58	R/W	0h	Select input qualification type for GPIO58: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-31. GPBQSEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	GPIO57	R/W	0h	Select input qualification type for GPIO57: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
17-16	GPIO56	R/W	0h	Select input qualification type for GPIO56: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
15-14	GPIO55	R/W	0h	Select input qualification type for GPIO55: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
13-12	GPIO54	R/W	0h	Select input qualification type for GPIO54: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
11-10	GPIO53	R/W	0h	Select input qualification type for GPIO53: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
9-8	GPIO52	R/W	0h	Select input qualification type for GPIO52: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
7-6	GPIO51	R/W	0h	Select input qualification type for GPIO51: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
5-4	GPIO50	R/W	0h	Select input qualification type for GPIO50: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	GPIO49	R/W	0h	Select input qualification type for GPIO49: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-31. GPBQSEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO48	R/W	0h	Select input qualification type for GPIO48: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

15.11.2.20 GPBMUX1 Register (Offset = 8Ch) [Reset = 00000000h]

GPBMUX1 is shown in Figure 15-23 and described in Table 15-32.

Return to the Summary Table.

GPIO B Mux 1 Register (GPIO32 to 47)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 15-23. GPBMUX1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO47		GPIO46		GPIO45		GPIO44		GPIO43		GPIO42		GPIO41		GPIO40	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO39		GPIO38		GPIO37		GPIO36		GPIO35		GPIO34		GPIO33		GPIO32	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-32. GPBMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO47	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO46	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO45	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO44	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO43	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO42	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO41	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO40	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO39	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO38	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO37	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO36	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO35	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO34	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO33	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-32. GPBMUX1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO32	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.21 GPBMUX2 Register (Offset = 90h) [Reset = 00000000h]

GPBMUX2 is shown in Figure 15-24 and described in Table 15-33.

Return to the Summary Table.

GPIO B Mux 2 Register (GPIO48 to 63)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 15-24. GPBMUX2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO63		GPIO62		GPIO61		GPIO60		GPIO59		GPIO58		GPIO57		GPIO56	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO55		GPIO54		GPIO53		GPIO52		GPIO51		GPIO50		GPIO49		GPIO48	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-33. GPBMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO63	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO62	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO61	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO60	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO59	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO58	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO57	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO56	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO55	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO54	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO53	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO52	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO51	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO50	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO49	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-33. GPBMUX2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO48	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.22 GPBPUD Register (Offset = 98h) [Reset = FFFFFFFFh]

GPBPUD is shown in Figure 15-25 and described in Table 15-34.

Return to the Summary Table.

GPIO B Pull Up Disable Register (GPIO32 to 63)
Disables the Pull-Up on GPIO.
0: Enables the Pull-Up.
1: Disables the Pull-Up.
Reading the register returns the current value of the register setting.
Note:
[1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low.

Figure 15-25. GPBPUD Register

31	30	29	28	27	26	25	24
GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 15-34. GPBPUD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO63	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
30	GPIO62	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
29	GPIO61	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
28	GPIO60	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
27	GPIO59	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
26	GPIO58	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
25	GPIO57	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
24	GPIO56	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
23	GPIO55	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
22	GPIO54	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
21	GPIO53	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn

Table 15-34. GPBPUD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO52	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
19	GPIO51	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
18	GPIO50	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
17	GPIO49	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
16	GPIO48	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
15	GPIO47	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
14	GPIO46	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
13	GPIO45	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
12	GPIO44	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
11	GPIO43	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
10	GPIO42	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
9	GPIO41	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
8	GPIO40	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
7	GPIO39	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
6	GPIO38	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
5	GPIO37	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
4	GPIO36	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
3	GPIO35	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
2	GPIO34	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
1	GPIO33	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
0	GPIO32	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn

15.11.2.23 GPBINV Register (Offset = A0h) [Reset = 00000000h]

GPBINV is shown in Figure 15-26 and described in Table 15-35.

Return to the Summary Table.

GPIO B Input Polarity Invert Registers (GPIO32 to 63)
Selects between non-inverted and inverted GPIO input to the device.
0: selects non-inverted GPIO input
1: selects inverted GPIO input
Reading the register returns the current value of the register setting.

Figure 15-26. GPBINV Register

31	30	29	28	27	26	25	24
GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-35. GPBINV Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO63	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
30	GPIO62	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
29	GPIO61	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
28	GPIO60	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
27	GPIO59	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
26	GPIO58	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
25	GPIO57	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
24	GPIO56	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
23	GPIO55	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
22	GPIO54	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
21	GPIO53	R/W	0h	Input inversion control for this pin Reset type: SYSRSn

Table 15-35. GPBINV Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO52	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
19	GPIO51	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
18	GPIO50	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
17	GPIO49	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
16	GPIO48	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
15	GPIO47	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
14	GPIO46	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
13	GPIO45	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
12	GPIO44	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
11	GPIO43	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
10	GPIO42	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
9	GPIO41	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
8	GPIO40	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
7	GPIO39	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
6	GPIO38	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
5	GPIO37	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
4	GPIO36	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
3	GPIO35	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
2	GPIO34	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
1	GPIO33	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
0	GPIO32	R/W	0h	Input inversion control for this pin Reset type: SYSRSn

15.11.2.24 GPBODR Register (Offset = A4h) [Reset = 00000000h]

GPBODR is shown in Figure 15-27 and described in Table 15-36.

Return to the Summary Table.

GPIO B Open Drain Output Register (GPIO32 to GPIO63)
Selects between normal and open-drain output for the GPIO pin.
0: Normal Output
1: Open Drain Output
Reading the register returns the current value of the register setting.
Note:
[1] In the Open Drain output mode, if the buffer is configured for output mode, a 0 value to be driven out comes out on the on the PAD while a 1 value to be driven out tri-states the buffer.

Figure 15-27. GPBODR Register

31	30	29	28	27	26	25	24
GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-36. GPBODR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO63	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
30	GPIO62	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
29	GPIO61	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
28	GPIO60	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
27	GPIO59	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
26	GPIO58	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
25	GPIO57	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
24	GPIO56	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
23	GPIO55	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
22	GPIO54	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn

Table 15-36. GPBODR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	GPIO53	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
20	GPIO52	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
19	GPIO51	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
18	GPIO50	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
17	GPIO49	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
16	GPIO48	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
15	GPIO47	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
14	GPIO46	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
13	GPIO45	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
12	GPIO44	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
11	GPIO43	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
10	GPIO42	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
9	GPIO41	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
8	GPIO40	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
7	GPIO39	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
6	GPIO38	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
5	GPIO37	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
4	GPIO36	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
3	GPIO35	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
2	GPIO34	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
1	GPIO33	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
0	GPIO32	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn

15.11.2.25 GPBGMUX1 Register (Offset = C0h) [Reset = 00000000h]

GPBGMUX1 is shown in Figure 15-28 and described in Table 15-37.

Return to the Summary Table.

GPIO B Peripheral Group Mux (GPIO32 to 47)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 15-28. GPBGMUX1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO47		GPIO46		GPIO45		GPIO44		GPIO43		GPIO42		GPIO41		GPIO40	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO39		GPIO38		GPIO37		GPIO36		GPIO35		GPIO34		GPIO33		GPIO32	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-37. GPBGMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO47	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO46	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO45	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO44	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO43	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO42	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO41	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO40	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO39	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO38	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO37	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO36	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO35	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO34	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO33	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-37. GPBGMUX1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO32	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.26 GPBGMUX2 Register (Offset = C4h) [Reset = 00000000h]

GPBGMUX2 is shown in Figure 15-29 and described in Table 15-38.

Return to the Summary Table.

GPIO B Peripheral Group Mux (GPIO48 to 63)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 15-29. GPBGMUX2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO63		GPIO62		GPIO61		GPIO60		GPIO59		GPIO58		GPIO57		GPIO56	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO55		GPIO54		GPIO53		GPIO52		GPIO51		GPIO50		GPIO49		GPIO48	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-38. GPBGMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO63	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO62	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO61	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO60	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO59	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO58	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO57	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO56	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO55	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO54	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO53	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO52	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO51	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO50	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO49	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-38. GPBGMUX2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO48	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.27 GPBCSEL1 Register (Offset = D0h) [Reset = 00000000h]

GPBCSEL1 is shown in Figure 15-30 and described in Table 15-39.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-30. GPBCSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO39				GPIO38				GPIO37				GPIO36			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO35				GPIO34				GPIO33				GPIO32			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-39. GPBCSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO39	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO38	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO37	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO36	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO35	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO34	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO33	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO32	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.28 GPBCSEL2 Register (Offset = D4h) [Reset = 00000000h]

GPBCSEL2 is shown in Figure 15-31 and described in Table 15-40.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-31. GPBCSEL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO47				GPIO46				GPIO45				GPIO44			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO43				GPIO42				GPIO41				GPIO40			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-40. GPBCSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO47	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO46	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO45	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO44	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO43	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO42	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO41	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO40	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.29 GPBCSEL3 Register (Offset = D8h) [Reset = 00000000h]

GPBCSEL3 is shown in Figure 15-32 and described in Table 15-41.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-32. GPBCSEL3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO55				GPIO54				GPIO53				GPIO52			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO51				GPIO50				GPIO49				GPIO48			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-41. GPBCSEL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO55	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO54	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO53	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO52	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO51	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO50	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO49	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO48	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.30 GPBCSEL4 Register (Offset = DCh) [Reset = 00000000h]

GPBCSEL4 is shown in Figure 15-33 and described in Table 15-42.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-33. GPBCSEL4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO63				GPIO62				GPIO61				GPIO60			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO59				GPIO58				GPIO57				GPIO56			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-42. GPBCSEL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO63	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO62	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO61	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO60	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO59	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO58	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO57	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO56	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.31 GPBLOCK Register (Offset = F8h) [Reset = 00000000h]

GPBLOCK is shown in Figure 15-34 and described in Table 15-43.

Return to the Summary Table.

GPIO B Lock Configuration Register (GPIO32 to 63)
GPIO Configuration Lock for GPIO.
0: Bits in GPyMUX1, GPyMUX2, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed
1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin

Figure 15-34. GPBLOCK Register

31	30	29	28	27	26	25	24
GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-43. GPBLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO63	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
30	GPIO62	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
29	GPIO61	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
28	GPIO60	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
27	GPIO59	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
26	GPIO58	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
25	GPIO57	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
24	GPIO56	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
23	GPIO55	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
22	GPIO54	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
21	GPIO53	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn

Table 15-43. GPBLOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO52	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
19	GPIO51	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
18	GPIO50	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
17	GPIO49	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
16	GPIO48	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
15	GPIO47	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
14	GPIO46	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
13	GPIO45	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
12	GPIO44	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
11	GPIO43	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
10	GPIO42	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
9	GPIO41	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
8	GPIO40	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
7	GPIO39	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
6	GPIO38	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
5	GPIO37	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
4	GPIO36	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
3	GPIO35	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
2	GPIO34	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
1	GPIO33	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
0	GPIO32	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn

15.11.2.32 GPBCR Register (Offset = FCh) [Reset = 00000000h]

GPBCR is shown in Figure 15-35 and described in Table 15-44.

Return to the Summary Table.

GPIO B Lock Commit Register (GPIO32 to 63)
GPIO Configuration Lock Commit for GPIO:
1: Locks changes to the bit in GPyLOCK register which controls the same pin
0: Bit in the GPyLOCK register which controls the same pin can be changed

Figure 15-35. GPBCR Register

31	30	29	28	27	26	25	24
GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 15-44. GPBCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO63	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
30	GPIO62	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
29	GPIO61	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
28	GPIO60	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
27	GPIO59	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
26	GPIO58	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
25	GPIO57	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
24	GPIO56	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
23	GPIO55	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
22	GPIO54	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
21	GPIO53	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
20	GPIO52	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn

Table 15-44. GPBCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO51	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
18	GPIO50	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
17	GPIO49	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
16	GPIO48	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
15	GPIO47	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
14	GPIO46	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
13	GPIO45	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
12	GPIO44	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
11	GPIO43	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
10	GPIO42	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
9	GPIO41	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
8	GPIO40	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
7	GPIO39	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
6	GPIO38	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
5	GPIO37	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
4	GPIO36	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
3	GPIO35	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
2	GPIO34	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
1	GPIO33	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
0	GPIO32	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn

15.11.2.33 GPCCTRL Register (Offset = 100h) [Reset = 00000000h]

GPCCTRL is shown in Figure 15-36 and described in Table 15-45.

Return to the Summary Table.

GPIO C Qualification Sampling Period Control (GPIO64 to 95)

Figure 15-36. GPCCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUALPRD3								QUALPRD2								QUALPRD1								QUALPRD0							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 15-45. GPCCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	QUALPRD3	R/W	0h	Qualification sampling period for GPIO88 to GPIO95: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
23-16	QUALPRD2	R/W	0h	Qualification sampling period for GPIO80 to GPIO87: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
15-8	QUALPRD1	R/W	0h	Qualification sampling period for GPIO72 to GPIO79: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
7-0	QUALPRD0	R/W	0h	Qualification sampling period for GPIO64 to GPIO71: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn

15.11.2.34 GPCQSEL1 Register (Offset = 104h) [Reset = 00000000h]

GPCQSEL1 is shown in Figure 15-37 and described in Table 15-46.

Return to the Summary Table.

GPIO C Qualifier Select 1 Register (GPIO64 to 79)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 15-37. GPCQSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO79		GPIO78		GPIO77		GPIO76		GPIO75		GPIO74		GPIO73		GPIO72	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO71		GPIO70		GPIO69		GPIO68		GPIO67		GPIO66		GPIO65		GPIO64	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-46. GPCQSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO79	R/W	0h	Select input qualification type for GPIO79: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
29-28	GPIO78	R/W	0h	Select input qualification type for GPIO78: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
27-26	GPIO77	R/W	0h	Select input qualification type for GPIO77: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
25-24	GPIO76	R/W	0h	Select input qualification type for GPIO76: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
23-22	GPIO75	R/W	0h	Select input qualification type for GPIO75: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
21-20	GPIO74	R/W	0h	Select input qualification type for GPIO74: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-46. GPCQSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	GPIO73	R/W	0h	Select input qualification type for GPIO73: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
17-16	GPIO72	R/W	0h	Select input qualification type for GPIO72: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
15-14	GPIO71	R/W	0h	Select input qualification type for GPIO71: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
13-12	GPIO70	R/W	0h	Select input qualification type for GPIO70: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
11-10	GPIO69	R/W	0h	Select input qualification type for GPIO69: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
9-8	GPIO68	R/W	0h	Select input qualification type for GPIO68: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
7-6	GPIO67	R/W	0h	Select input qualification type for GPIO67: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
5-4	GPIO66	R/W	0h	Select input qualification type for GPIO66: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	GPIO65	R/W	0h	Select input qualification type for GPIO65: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-46. GPCQSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO64	R/W	0h	Select input qualification type for GPIO64: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

15.11.2.35 GPCQSEL2 Register (Offset = 108h) [Reset = 00000000h]

GPCQSEL2 is shown in Figure 15-38 and described in Table 15-47.

Return to the Summary Table.

GPIO C Qualifier Select 2 Register (GPIO80 to 95)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 15-38. GPCQSEL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO95		GPIO94		GPIO93		GPIO92		GPIO91		GPIO90		GPIO89		GPIO88	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO87		GPIO86		GPIO85		GPIO84		GPIO83		GPIO82		GPIO81		GPIO80	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-47. GPCQSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO95	R/W	0h	Select input qualification type for GPIO95: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
29-28	GPIO94	R/W	0h	Select input qualification type for GPIO94: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
27-26	GPIO93	R/W	0h	Select input qualification type for GPIO93: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
25-24	GPIO92	R/W	0h	Select input qualification type for GPIO92: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
23-22	GPIO91	R/W	0h	Select input qualification type for GPIO91: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
21-20	GPIO90	R/W	0h	Select input qualification type for GPIO90: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-47. GPCQSEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	GPIO89	R/W	0h	Select input qualification type for GPIO89: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
17-16	GPIO88	R/W	0h	Select input qualification type for GPIO88: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
15-14	GPIO87	R/W	0h	Select input qualification type for GPIO87: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
13-12	GPIO86	R/W	0h	Select input qualification type for GPIO86: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
11-10	GPIO85	R/W	0h	Select input qualification type for GPIO85: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
9-8	GPIO84	R/W	0h	Select input qualification type for GPIO84: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
7-6	GPIO83	R/W	0h	Select input qualification type for GPIO83: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
5-4	GPIO82	R/W	0h	Select input qualification type for GPIO82: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	GPIO81	R/W	0h	Select input qualification type for GPIO81: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-47. GPCQSEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO80	R/W	0h	Select input qualification type for GPIO80: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

15.11.2.36 GPCMUX1 Register (Offset = 10Ch) [Reset = 00000000h]

GPCMUX1 is shown in Figure 15-39 and described in Table 15-48.

Return to the Summary Table.

GPIO C Mux 1 Register (GPIO64 to 79)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 15-39. GPCMUX1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO79		GPIO78		GPIO77		GPIO76		GPIO75		GPIO74		GPIO73		GPIO72	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO71		GPIO70		GPIO69		GPIO68		GPIO67		GPIO66		GPIO65		GPIO64	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-48. GPCMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO79	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO78	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO77	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO76	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO75	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO74	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO73	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO72	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO71	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO70	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO69	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO68	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO67	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO66	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO65	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-48. GPCMUX1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO64	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.37 GPCMUX2 Register (Offset = 110h) [Reset = 00000000h]

GPCMUX2 is shown in Figure 15-40 and described in Table 15-49.

Return to the Summary Table.

GPIO C Mux 2 Register (GPIO80 to 95)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 15-40. GPCMUX2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO95		GPIO94		GPIO93		GPIO92		GPIO91		GPIO90		GPIO89		GPIO88	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO87		GPIO86		GPIO85		GPIO84		GPIO83		GPIO82		GPIO81		GPIO80	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-49. GPCMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO95	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO94	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO93	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO92	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO91	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO90	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO89	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO88	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO87	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO86	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO85	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO84	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO83	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO82	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO81	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-49. GPCMUX2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO80	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.38 GPCPUD Register (Offset = 118h) [Reset = FFFFFFFFh]

GPCPUD is shown in Figure 15-41 and described in Table 15-50.

Return to the Summary Table.

GPIO C Pull Up Disable Register (GPIO64 to 95)
Disables the Pull-Up on GPIO.
0: Enables the Pull-Up.
1: Disables the Pull-Up.
Reading the register returns the current value of the register setting.
Note:
[1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low.

Figure 15-41. GPCPUD Register

31	30	29	28	27	26	25	24
GPIO95	GPIO94	GPIO93	GPIO92	GPIO91	GPIO90	GPIO89	GPIO88
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
GPIO87	GPIO86	GPIO85	GPIO84	GPIO83	GPIO82	GPIO81	GPIO80
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
GPIO79	GPIO78	GPIO77	GPIO76	GPIO75	GPIO74	GPIO73	GPIO72
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 15-50. GPCPUD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO95	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
30	GPIO94	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
29	GPIO93	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
28	GPIO92	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
27	GPIO91	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
26	GPIO90	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
25	GPIO89	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
24	GPIO88	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
23	GPIO87	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
22	GPIO86	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
21	GPIO85	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn

Table 15-50. GPCPUD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO84	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
19	GPIO83	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
18	GPIO82	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
17	GPIO81	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
16	GPIO80	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
15	GPIO79	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
14	GPIO78	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
13	GPIO77	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
12	GPIO76	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
11	GPIO75	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
10	GPIO74	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
9	GPIO73	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
8	GPIO72	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
7	GPIO71	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
6	GPIO70	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
5	GPIO69	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
4	GPIO68	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
3	GPIO67	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
2	GPIO66	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
1	GPIO65	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
0	GPIO64	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn

15.11.2.39 GPCINV Register (Offset = 120h) [Reset = 00000000h]

GPCINV is shown in Figure 15-42 and described in Table 15-51.

Return to the Summary Table.

GPIO C Input Polarity Invert Registers (GPIO64 to 95)
Selects between non-inverted and inverted GPIO input to the device.
0: selects non-inverted GPIO input
1: selects inverted GPIO input
Reading the register returns the current value of the register setting.

Figure 15-42. GPCINV Register

31	30	29	28	27	26	25	24
GPIO95	GPIO94	GPIO93	GPIO92	GPIO91	GPIO90	GPIO89	GPIO88
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO87	GPIO86	GPIO85	GPIO84	GPIO83	GPIO82	GPIO81	GPIO80
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO79	GPIO78	GPIO77	GPIO76	GPIO75	GPIO74	GPIO73	GPIO72
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-51. GPCINV Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO95	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
30	GPIO94	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
29	GPIO93	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
28	GPIO92	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
27	GPIO91	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
26	GPIO90	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
25	GPIO89	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
24	GPIO88	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
23	GPIO87	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
22	GPIO86	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
21	GPIO85	R/W	0h	Input inversion control for this pin Reset type: SYSRSn

Table 15-51. GPCINV Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO84	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
19	GPIO83	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
18	GPIO82	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
17	GPIO81	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
16	GPIO80	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
15	GPIO79	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
14	GPIO78	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
13	GPIO77	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
12	GPIO76	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
11	GPIO75	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
10	GPIO74	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
9	GPIO73	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
8	GPIO72	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
7	GPIO71	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
6	GPIO70	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
5	GPIO69	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
4	GPIO68	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
3	GPIO67	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
2	GPIO66	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
1	GPIO65	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
0	GPIO64	R/W	0h	Input inversion control for this pin Reset type: SYSRSn

15.11.2.40 GPCODR Register (Offset = 124h) [Reset = 00000000h]

GPCODR is shown in Figure 15-43 and described in Table 15-52.

Return to the Summary Table.

GPIO C Open Drain Output Register (GPIO64 to GPIO95)
Selects between normal and open-drain output for the GPIO pin.
0: Normal Output
1: Open Drain Output
Reading the register returns the current value of the register setting.
Note:
[1] In the Open Drain output mode, if the buffer is configured for output mode, a 0 value to be driven out comes out on the on the PAD while a 1 value to be driven out tri-states the buffer.

Figure 15-43. GPCODR Register

31	30	29	28	27	26	25	24
GPIO95	GPIO94	GPIO93	GPIO92	GPIO91	GPIO90	GPIO89	GPIO88
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO87	GPIO86	GPIO85	GPIO84	GPIO83	GPIO82	GPIO81	GPIO80
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO79	GPIO78	GPIO77	GPIO76	GPIO75	GPIO74	GPIO73	GPIO72
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-52. GPCODR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO95	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
30	GPIO94	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
29	GPIO93	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
28	GPIO92	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
27	GPIO91	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
26	GPIO90	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
25	GPIO89	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
24	GPIO88	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
23	GPIO87	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
22	GPIO86	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn

Table 15-52. GPCODR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	GPIO85	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
20	GPIO84	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
19	GPIO83	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
18	GPIO82	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
17	GPIO81	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
16	GPIO80	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
15	GPIO79	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
14	GPIO78	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
13	GPIO77	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
12	GPIO76	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
11	GPIO75	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
10	GPIO74	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
9	GPIO73	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
8	GPIO72	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
7	GPIO71	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
6	GPIO70	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
5	GPIO69	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
4	GPIO68	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
3	GPIO67	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
2	GPIO66	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
1	GPIO65	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
0	GPIO64	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn

15.11.2.41 GPCGMUX1 Register (Offset = 140h) [Reset = 00000000h]

GPCGMUX1 is shown in Figure 15-44 and described in Table 15-53.

Return to the Summary Table.

GPIO C Peripheral Group Mux (GPIO64 to 79)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 15-44. GPCGMUX1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO79		GPIO78		GPIO77		GPIO76		GPIO75		GPIO74		GPIO73		GPIO72	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO71		GPIO70		GPIO69		GPIO68		GPIO67		GPIO66		GPIO65		GPIO64	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-53. GPCGMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO79	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO78	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO77	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO76	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO75	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO74	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO73	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO72	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO71	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO70	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO69	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO68	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO67	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO66	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO65	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-53. GPCGMUX1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO64	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.42 GPCGMUX2 Register (Offset = 144h) [Reset = 00000000h]

GPCGMUX2 is shown in Figure 15-45 and described in Table 15-54.

Return to the Summary Table.

GPIO C Peripheral Group Mux (GPIO80 to 95)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 15-45. GPCGMUX2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO95		GPIO94		GPIO93		GPIO92		GPIO91		GPIO90		GPIO89		GPIO88	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO87		GPIO86		GPIO85		GPIO84		GPIO83		GPIO82		GPIO81		GPIO80	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-54. GPCGMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO95	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO94	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO93	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO92	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO91	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO90	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO89	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO88	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO87	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO86	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO85	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO84	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO83	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO82	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO81	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-54. GPCGMUX2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO80	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.43 GPCCSEL1 Register (Offset = 150h) [Reset = 00000000h]

GPCCSEL1 is shown in Figure 15-46 and described in Table 15-55.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-46. GPCCSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO71				GPIO70				GPIO69				GPIO68			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO67				GPIO66				GPIO65				GPIO64			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-55. GPCCSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO71	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO70	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO69	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO68	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO67	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO66	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO65	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO64	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.44 GPCCSEL2 Register (Offset = 154h) [Reset = 00000000h]

GPCCSEL2 is shown in Figure 15-47 and described in Table 15-56.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-47. GPCCSEL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO79				GPIO78				GPIO77				GPIO76			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO75				GPIO74				GPIO73				GPIO72			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-56. GPCCSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO79	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO78	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO77	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO76	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO75	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO74	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO73	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO72	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.45 GPCCSEL3 Register (Offset = 158h) [Reset = 00000000h]

GPCCSEL3 is shown in Figure 15-48 and described in Table 15-57.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-48. GPCCSEL3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO87				GPIO86				GPIO85				GPIO84			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO83				GPIO82				GPIO81				GPIO80			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-57. GPCCSEL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO87	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO86	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO85	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO84	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO83	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO82	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO81	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO80	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.46 GPCCSEL4 Register (Offset = 15Ch) [Reset = 00000000h]

GPCCSEL4 is shown in Figure 15-49 and described in Table 15-58.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-49. GPCCSEL4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO95				GPIO94				GPIO93				GPIO92			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO91				GPIO90				GPIO89				GPIO88			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-58. GPCCSEL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO95	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO94	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO93	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO92	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO91	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO90	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO89	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO88	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.47 GPCLOCK Register (Offset = 178h) [Reset = 00000000h]

GPCLOCK is shown in Figure 15-50 and described in Table 15-59.

Return to the Summary Table.

GPIO C Lock Configuration Register (GPIO64 to 95)
GPIO Configuration Lock for GPIO.
0: Bits in GPyMUX1, GPyMUX2, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed
1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin

Figure 15-50. GPCLOCK Register

31	30	29	28	27	26	25	24
GPIO95	GPIO94	GPIO93	GPIO92	GPIO91	GPIO90	GPIO89	GPIO88
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO87	GPIO86	GPIO85	GPIO84	GPIO83	GPIO82	GPIO81	GPIO80
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO79	GPIO78	GPIO77	GPIO76	GPIO75	GPIO74	GPIO73	GPIO72
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-59. GPCLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO95	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
30	GPIO94	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
29	GPIO93	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
28	GPIO92	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
27	GPIO91	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
26	GPIO90	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
25	GPIO89	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
24	GPIO88	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
23	GPIO87	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
22	GPIO86	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
21	GPIO85	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn

Table 15-59. GPCLOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO84	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
19	GPIO83	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
18	GPIO82	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
17	GPIO81	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
16	GPIO80	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
15	GPIO79	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
14	GPIO78	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
13	GPIO77	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
12	GPIO76	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
11	GPIO75	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
10	GPIO74	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
9	GPIO73	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
8	GPIO72	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
7	GPIO71	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
6	GPIO70	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
5	GPIO69	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
4	GPIO68	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
3	GPIO67	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
2	GPIO66	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
1	GPIO65	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
0	GPIO64	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn

15.11.2.48 GPCCR Register (Offset = 17Ch) [Reset = 00000000h]

GPCCR is shown in Figure 15-51 and described in Table 15-60.

Return to the Summary Table.

GPIO C Lock Commit Register (GPIO64 to 95)
GPIO Configuration Lock Commit for GPIO:
1: Locks changes to the bit in GPyLOCK register which controls the same pin
0: Bit in the GPyLOCK register which controls the same pin can be changed

Figure 15-51. GPCCR Register

31	30	29	28	27	26	25	24
GPIO95	GPIO94	GPIO93	GPIO92	GPIO91	GPIO90	GPIO89	GPIO88
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
GPIO87	GPIO86	GPIO85	GPIO84	GPIO83	GPIO82	GPIO81	GPIO80
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
GPIO79	GPIO78	GPIO77	GPIO76	GPIO75	GPIO74	GPIO73	GPIO72
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 15-60. GPCCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO95	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
30	GPIO94	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
29	GPIO93	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
28	GPIO92	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
27	GPIO91	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
26	GPIO90	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
25	GPIO89	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
24	GPIO88	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
23	GPIO87	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
22	GPIO86	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
21	GPIO85	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
20	GPIO84	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn

Table 15-60. GPCCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO83	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
18	GPIO82	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
17	GPIO81	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
16	GPIO80	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
15	GPIO79	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
14	GPIO78	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
13	GPIO77	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
12	GPIO76	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
11	GPIO75	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
10	GPIO74	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
9	GPIO73	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
8	GPIO72	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
7	GPIO71	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
6	GPIO70	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
5	GPIO69	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
4	GPIO68	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
3	GPIO67	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
2	GPIO66	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
1	GPIO65	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
0	GPIO64	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn

15.11.2.49 GPDCTRL Register (Offset = 180h) [Reset = 00000000h]

GPDCTRL is shown in Figure 15-52 and described in Table 15-61.

Return to the Summary Table.

GPIO D Qualification Sampling Period Control (GPIO96 to 127)

Figure 15-52. GPDCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUALPRD3								RESERVED								QUALPRD1								QUALPRD0							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 15-61. GPDCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	QUALPRD3	R/W	0h	Qualification sampling period for GPIO120 to GPIO127: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
23-16	RESERVED	R/W	0h	Reserved
15-8	QUALPRD1	R/W	0h	Qualification sampling period for GPIO104 to GPIO111: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
7-0	QUALPRD0	R/W	0h	Qualification sampling period for GPIO96 to GPIO103: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn

15.11.2.50 GPDQSEL1 Register (Offset = 184h) [Reset = 00000000h]

GPDQSEL1 is shown in Figure 15-53 and described in Table 15-62.

Return to the Summary Table.

GPIO D Qualifier Select 1 Register (GPIO96 to 111)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 15-53. GPDQSEL1 Register

31	30	29	28	27	26	25	24
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		RESERVED		GPIO105		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO103		RESERVED		GPIO101		GPIO100	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO99		GPIO98		GPIO97		GPIO96	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-62. GPDQSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	GPIO105	R/W	0h	Select input qualification type for GPIO105: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
17-16	RESERVED	R/W	0h	Reserved
15-14	GPIO103	R/W	0h	Select input qualification type for GPIO103: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
13-12	RESERVED	R/W	0h	Reserved

Table 15-62. GPDQSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-10	GPIO101	R/W	0h	Select input qualification type for GPIO101: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
9-8	GPIO100	R/W	0h	Select input qualification type for GPIO100: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
7-6	GPIO99	R/W	0h	Select input qualification type for GPIO99: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
5-4	GPIO98	R/W	0h	Select input qualification type for GPIO98: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	GPIO97	R/W	0h	Select input qualification type for GPIO97: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
1-0	GPIO96	R/W	0h	Select input qualification type for GPIO96: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

15.11.2.51 GPDQSEL2 Register (Offset = 188h) [Reset = 00000000h]

GPDQSEL2 is shown in Figure 15-54 and described in Table 15-63.

Return to the Summary Table.

GPIO D Qualifier Select 2 Register (GPIO112 to 127)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 15-54. GPDQSEL2 Register

31	30	29	28	27	26	25	24
GPIO127		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-63. GPDQSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO127	R/W	0h	Select input qualification type for GPIO127: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3-2	RESERVED	R/W	0h	Reserved
1-0	RESERVED	R/W	0h	Reserved

15.11.2.52 GPDMUX1 Register (Offset = 18Ch) [Reset = 00000000h]

GPDMUX1 is shown in Figure 15-55 and described in Table 15-64.

Return to the Summary Table.

GPIO D Mux 1 Register (GPIO96 to 111)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 15-55. GPDMUX1 Register

31	30	29	28	27	26	25	24
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		RESERVED		GPIO105		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO103		RESERVED		GPIO101		GPIO100	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO99		GPIO98		GPIO97		GPIO96	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-64. GPDMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	GPIO105	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	RESERVED	R/W	0h	Reserved
15-14	GPIO103	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	RESERVED	R/W	0h	Reserved
11-10	GPIO101	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO100	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO99	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO98	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO97	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-64. GPDMUX1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO96	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.53 GPDMUX2 Register (Offset = 190h) [Reset = 00000000h]

GPDMUX2 is shown in Figure 15-56 and described in Table 15-65.

Return to the Summary Table.

GPIO D Mux 2 Register (GPIO112 to 127)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 15-56. GPDMUX2 Register

31	30	29	28	27	26	25	24
GPIO127		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-65. GPDMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO127	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3-2	RESERVED	R/W	0h	Reserved
1-0	RESERVED	R/W	0h	Reserved

15.11.2.54 GPDPUD Register (Offset = 198h) [Reset = FFFFFFFFh]

GPDPUD is shown in Figure 15-57 and described in Table 15-66.

Return to the Summary Table.

GPIO D Pull Up Disable Register (GPIO96 to 127)
Disables the Pull-Up on GPIO.
0: Enables the Pull-Up.
1: Disables the Pull-Up.
Reading the register returns the current value of the register setting.
Note:
[1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low.

Figure 15-57. GPDPUD Register

31	30	29	28	27	26	25	24
GPIO127	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO105	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
GPIO103	RESERVED	GPIO101	GPIO100	GPIO99	GPIO98	GPIO97	GPIO96
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 15-66. GPDPUD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO127	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
30	RESERVED	R/W	1h	Reserved
29	RESERVED	R/W	1h	Reserved
28	RESERVED	R/W	1h	Reserved
27	RESERVED	R/W	1h	Reserved
26	RESERVED	R/W	1h	Reserved
25	RESERVED	R/W	1h	Reserved
24	RESERVED	R/W	1h	Reserved
23	RESERVED	R/W	1h	Reserved
22	RESERVED	R/W	1h	Reserved
21	RESERVED	R/W	1h	Reserved
20	RESERVED	R/W	1h	Reserved
19	RESERVED	R/W	1h	Reserved
18	RESERVED	R/W	1h	Reserved
17	RESERVED	R/W	1h	Reserved
16	RESERVED	R/W	1h	Reserved
15	RESERVED	R/W	1h	Reserved
14	RESERVED	R/W	1h	Reserved
13	RESERVED	R/W	1h	Reserved

Table 15-66. GPDPUD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	RESERVED	R/W	1h	Reserved
11	RESERVED	R/W	1h	Reserved
10	RESERVED	R/W	1h	Reserved
9	GPIO105	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
8	RESERVED	R/W	1h	Reserved
7	GPIO103	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
6	RESERVED	R/W	1h	Reserved
5	GPIO101	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
4	GPIO100	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
3	GPIO99	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
2	GPIO98	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
1	GPIO97	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
0	GPIO96	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn

15.11.2.55 GPDINV Register (Offset = 1A0h) [Reset = 00000000h]

GPDINV is shown in Figure 15-58 and described in Table 15-67.

Return to the Summary Table.

GPIO D Input Polarity Invert Registers (GPIO96 to 127)
Selects between non-inverted and inverted GPIO input to the device.
0: selects non-inverted GPIO input
1: selects inverted GPIO input
Reading the register returns the current value of the register setting.

Figure 15-58. GPDINV Register

31	30	29	28	27	26	25	24
GPIO127	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO105	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO103	RESERVED	GPIO101	GPIO100	GPIO99	GPIO98	GPIO97	GPIO96
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-67. GPDINV Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO127	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved

Table 15-67. GPDINV Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RESERVED	R/W	0h	Reserved
9	GPIO105	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
8	RESERVED	R/W	0h	Reserved
7	GPIO103	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
6	RESERVED	R/W	0h	Reserved
5	GPIO101	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
4	GPIO100	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
3	GPIO99	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
2	GPIO98	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
1	GPIO97	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
0	GPIO96	R/W	0h	Input inversion control for this pin Reset type: SYSRSn

15.11.2.56 GPDODR Register (Offset = 1A4h) [Reset = 00000000h]

GPDODR is shown in Figure 15-59 and described in Table 15-68.

Return to the Summary Table.

GPIO D Open Drain Output Register (GPIO96 to GPIO127)
Selects between normal and open-drain output for the GPIO pin.
0: Normal Output
1: Open Drain Output
Reading the register returns the current value of the register setting.
Note:
[1] In the Open Drain output mode, if the buffer is configured for output mode, a 0 value to be driven out comes out on the on the PAD while a 1 value to be driven out tri-states the buffer.

Figure 15-59. GPDODR Register

31	30	29	28	27	26	25	24
GPIO127	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO105	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO103	RESERVED	GPIO101	GPIO100	GPIO99	GPIO98	GPIO97	GPIO96
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-68. GPDODR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO127	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved

Table 15-68. GPDODR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	GPIO105	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
8	RESERVED	R/W	0h	Reserved
7	GPIO103	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
6	RESERVED	R/W	0h	Reserved
5	GPIO101	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
4	GPIO100	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
3	GPIO99	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
2	GPIO98	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
1	GPIO97	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
0	GPIO96	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn

15.11.2.57 GPDGMUX1 Register (Offset = 1C0h) [Reset = 00000000h]

GPDGMUX1 is shown in Figure 15-60 and described in Table 15-69.

Return to the Summary Table.

GPIO D Peripheral Group Mux (GPIO96 to 111)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 15-60. GPDGMUX1 Register

31	30	29	28	27	26	25	24
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		RESERVED		GPIO105		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO103		RESERVED		GPIO101		GPIO100	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO99		GPIO98		GPIO97		GPIO96	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-69. GPDGMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	GPIO105	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	RESERVED	R/W	0h	Reserved
15-14	GPIO103	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	RESERVED	R/W	0h	Reserved
11-10	GPIO101	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO100	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO99	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO98	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO97	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-69. GPDGMUX1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO96	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.58 GPDGMUX2 Register (Offset = 1C4h) [Reset = 00000000h]

GPDGMUX2 is shown in Figure 15-61 and described in Table 15-70.

Return to the Summary Table.

GPIO D Peripheral Group Mux (GPIO112 to 127)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 15-61. GPDGMUX2 Register

31	30	29	28	27	26	25	24
GPIO127		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-70. GPDGMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO127	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3-2	RESERVED	R/W	0h	Reserved
1-0	RESERVED	R/W	0h	Reserved

15.11.2.59 GPDCSEL1 Register (Offset = 1D0h) [Reset = 00000000h]

GPDCSEL1 is shown in Figure 15-62 and described in Table 15-71.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-62. GPDCSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO103				RESERVED				GPIO101				GPIO100			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO99				GPIO98				GPIO97				GPIO96			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-71. GPDCSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO103	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	RESERVED	R/W	0h	Reserved
23-20	GPIO101	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO100	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO99	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO98	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO97	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO96	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.60 GPDCSEL2 Register (Offset = 1D4h) [Reset = 00000000h]

GPDCSEL2 is shown in Figure 15-63 and described in Table 15-72.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-63. GPDCSEL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				RESERVED				RESERVED				RESERVED			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				RESERVED				GPIO105				RESERVED			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-72. GPDCSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	Reserved
27-24	RESERVED	R/W	0h	Reserved
23-20	RESERVED	R/W	0h	Reserved
19-16	RESERVED	R/W	0h	Reserved
15-12	RESERVED	R/W	0h	Reserved
11-8	RESERVED	R/W	0h	Reserved
7-4	GPIO105	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	RESERVED	R/W	0h	Reserved

15.11.2.61 GPDCSEL3 Register (Offset = 1D8h) [Reset = 00000000h]

GPDCSEL3 is shown in Figure 15-64 and described in Table 15-73.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-64. GPDCSEL3 Register

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

Table 15-73. GPDCSEL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	Reserved
27-24	RESERVED	R/W	0h	Reserved
23-20	RESERVED	R/W	0h	Reserved
19-16	RESERVED	R/W	0h	Reserved
15-12	RESERVED	R/W	0h	Reserved
11-8	RESERVED	R/W	0h	Reserved
7-4	RESERVED	R/W	0h	Reserved
3-0	RESERVED	R/W	0h	Reserved

15.11.2.62 GPDCSEL4 Register (Offset = 1DCh) [Reset = 00000000h]

GPDCSEL4 is shown in Figure 15-65 and described in Table 15-74.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-65. GPDCSEL4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO127				RESERVED				RESERVED				RESERVED			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				RESERVED				RESERVED				RESERVED			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-74. GPDCSEL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO127	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	RESERVED	R/W	0h	Reserved
23-20	RESERVED	R/W	0h	Reserved
19-16	RESERVED	R/W	0h	Reserved
15-12	RESERVED	R/W	0h	Reserved
11-8	RESERVED	R/W	0h	Reserved
7-4	RESERVED	R/W	0h	Reserved
3-0	RESERVED	R/W	0h	Reserved

15.11.2.63 GPDLOCK Register (Offset = 1F8h) [Reset = 00000000h]

GPDLOCK is shown in Figure 15-66 and described in Table 15-75.

Return to the Summary Table.

GPIO D Lock Configuration Register (GPIO96 to 127)
GPIO Configuration Lock for GPIO.
0: Bits in GPyMUX1, GPyMUX2, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed
1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin

Figure 15-66. GPDLOCK Register

31	30	29	28	27	26	25	24
GPIO127	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO105	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO103	RESERVED	GPIO101	GPIO100	GPIO99	GPIO98	GPIO97	GPIO96
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-75. GPDLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO127	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved

Table 15-75. GPDLOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	GPIO105	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
8	RESERVED	R/W	0h	Reserved
7	GPIO103	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
6	RESERVED	R/W	0h	Reserved
5	GPIO101	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
4	GPIO100	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
3	GPIO99	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
2	GPIO98	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
1	GPIO97	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
0	GPIO96	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn

15.11.2.64 GPDCR Register (Offset = 1FCh) [Reset = 00000000h]

GPDCR is shown in Figure 15-67 and described in Table 15-76.

Return to the Summary Table.

GPIO D Lock Commit Register (GPIO96 to 127)
GPIO Configuration Lock Commit for GPIO:
1: Locks changes to the bit in GPyLOCK register which controls the same pin
0: Bit in the GPyLOCK register which controls the same pin can be changed

Figure 15-67. GPDCR Register

31	30	29	28	27	26	25	24
GPIO127	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO105	RESERVED
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
GPIO103	RESERVED	GPIO101	GPIO100	GPIO99	GPIO98	GPIO97	GPIO96
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 15-76. GPDCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO127	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
30	RESERVED	R/WSonce	0h	Reserved
29	RESERVED	R/WSonce	0h	Reserved
28	RESERVED	R/WSonce	0h	Reserved
27	RESERVED	R/WSonce	0h	Reserved
26	RESERVED	R/WSonce	0h	Reserved
25	RESERVED	R/WSonce	0h	Reserved
24	RESERVED	R/WSonce	0h	Reserved
23	RESERVED	R/WSonce	0h	Reserved
22	RESERVED	R/WSonce	0h	Reserved
21	RESERVED	R/WSonce	0h	Reserved
20	RESERVED	R/WSonce	0h	Reserved
19	RESERVED	R/WSonce	0h	Reserved
18	RESERVED	R/WSonce	0h	Reserved
17	RESERVED	R/WSonce	0h	Reserved
16	RESERVED	R/WSonce	0h	Reserved
15	RESERVED	R/WSonce	0h	Reserved
14	RESERVED	R/WSonce	0h	Reserved
13	RESERVED	R/WSonce	0h	Reserved
12	RESERVED	R/WSonce	0h	Reserved
11	RESERVED	R/WSonce	0h	Reserved
10	RESERVED	R/WSonce	0h	Reserved

Table 15-76. GPDCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	GPIO105	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
8	RESERVED	R/WSonce	0h	Reserved
7	GPIO103	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
6	RESERVED	R/WSonce	0h	Reserved
5	GPIO101	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
4	GPIO100	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
3	GPIO99	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
2	GPIO98	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
1	GPIO97	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
0	GPIO96	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn

15.11.2.65 GPFCTRL Register (Offset = 280h) [Reset = 00000000h]

GPFCTRL is shown in Figure 15-68 and described in Table 15-77.

Return to the Summary Table.

GPIO F Qualification Sampling Period Control (GPIO160 to 191)

Figure 15-68. GPFCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUALPRD3								QUALPRD2								QUALPRD1								QUALPRD0							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 15-77. GPFCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	QUALPRD3	R/W	0h	Qualification sampling period for GPIO184 to GPIO191: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
23-16	QUALPRD2	R/W	0h	Qualification sampling period for GPIO176 to GPIO183: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
15-8	QUALPRD1	R/W	0h	Qualification sampling period for GPIO168: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
7-0	QUALPRD0	R/W	0h	Qualification sampling period for GPIO160 to GPIO167: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn

15.11.2.66 GPFQSEL1 Register (Offset = 284h) [Reset = 00000000h]

GPFQSEL1 is shown in Figure 15-69 and described in Table 15-78.

Return to the Summary Table.

GPIO F Qualifier Select 1 Register (GPIO160 to 168)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 15-69. GPFQSEL1 Register

31	30	29	28	27	26	25	24
GPIO175		GPIO174		GPIO173		GPIO172	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
GPIO171		GPIO170		GPIO169		GPIO168	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO167		GPIO166		GPIO165		GPIO164	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO163		GPIO162		GPIO161		GPIO160	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-78. GPFQSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO175	R/W	0h	Select input qualification type for GPIO175: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
29-28	GPIO174	R/W	0h	Select input qualification type for GPIO174: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
27-26	GPIO173	R/W	0h	Select input qualification type for GPIO173: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
25-24	GPIO172	R/W	0h	Select input qualification type for GPIO172: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-78. GPFQSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-22	GPIO171	R/W	0h	Select input qualification type for GPIO171: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
21-20	GPIO170	R/W	0h	Select input qualification type for GPIO170: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
19-18	GPIO169	R/W	0h	Select input qualification type for GPIO169: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
17-16	GPIO168	R/W	0h	Select input qualification type for GPIO168: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
15-14	GPIO167	R/W	0h	Select input qualification type for GPIO167: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
13-12	GPIO166	R/W	0h	Select input qualification type for GPIO166: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
11-10	GPIO165	R/W	0h	Select input qualification type for GPIO165: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
9-8	GPIO164	R/W	0h	Select input qualification type for GPIO164: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
7-6	GPIO163	R/W	0h	Select input qualification type for GPIO163: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-78. GPFQSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	GPIO162	R/W	0h	Select input qualification type for GPIO162: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	GPIO161	R/W	0h	Select input qualification type for GPIO161: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
1-0	GPIO160	R/W	0h	Select input qualification type for GPIO160: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

15.11.2.67 GPFQSEL2 Register (Offset = 288h) [Reset = 00000000h]

GPFQSEL2 is shown in Figure 15-70 and described in Table 15-79.

Return to the Summary Table.

GPIO F Qualifier Select 2 Register (GPIO176 to 191)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 15-70. GPFQSEL2 Register

31	30	29	28	27	26	25	24
GPIO191		GPIO190		GPIO189		GPIO188	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
GPIO187		GPIO186		GPIO185		GPIO184	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO183		GPIO182		GPIO181		GPIO180	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO179		GPIO178		GPIO177		GPIO176	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-79. GPFQSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO191	R/W	0h	Select input qualification type for GPIO191: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
29-28	GPIO190	R/W	0h	Select input qualification type for GPIO190: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
27-26	GPIO189	R/W	0h	Select input qualification type for GPIO189: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
25-24	GPIO188	R/W	0h	Select input qualification type for GPIO188: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-79. GPFQSEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-22	GPIO187	R/W	0h	Select input qualification type for GPIO187: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
21-20	GPIO186	R/W	0h	Select input qualification type for GPIO186: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
19-18	GPIO185	R/W	0h	Select input qualification type for GPIO185: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
17-16	GPIO184	R/W	0h	Select input qualification type for GPIO184: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
15-14	GPIO183	R/W	0h	Select input qualification type for GPIO183: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
13-12	GPIO182	R/W	0h	Select input qualification type for GPIO182: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
11-10	GPIO181	R/W	0h	Select input qualification type for GPIO181: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
9-8	GPIO180	R/W	0h	Select input qualification type for GPIO180: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
7-6	GPIO179	R/W	0h	Select input qualification type for GPIO179: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-79. GPFQSEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	GPIO178	R/W	0h	Select input qualification type for GPIO178: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	GPIO177	R/W	0h	Select input qualification type for GPIO177: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
1-0	GPIO176	R/W	0h	Select input qualification type for GPIO176: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

15.11.2.68 GPFMUX1 Register (Offset = 28Ch) [Reset = 00000000h]

GPFMUX1 is shown in Figure 15-71 and described in Table 15-80.

Return to the Summary Table.

GPIO F Mux 1 Register (GPIO160 to 175)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 15-71. GPFMUX1 Register

31	30	29	28	27	26	25	24
GPIO175		GPIO174		GPIO173		GPIO172	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
GPIO171		GPIO170		GPIO169		GPIO168	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO167		GPIO166		GPIO165		GPIO164	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO163		GPIO162		GPIO161		GPIO160	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-80. GPFMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO175	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO174	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO173	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO172	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO171	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO170	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO169	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO168	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO167	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO166	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO165	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-80. GPFMUX1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9-8	GPIO164	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO163	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO162	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO161	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
1-0	GPIO160	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.69 GPFMUX2 Register (Offset = 290h) [Reset = 00000000h]

GPFMUX2 is shown in Figure 15-72 and described in Table 15-81.

Return to the Summary Table.

GPIO F Mux 2 Register (GPIO176 to 191)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 15-72. GPFMUX2 Register

31	30	29	28	27	26	25	24
GPIO191		GPIO190		GPIO189		GPIO188	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
GPIO187		GPIO186		GPIO185		GPIO184	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO183		GPIO182		GPIO181		GPIO180	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO179		GPIO178		GPIO177		GPIO176	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-81. GPFMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO191	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO190	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO189	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO188	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO187	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO186	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO185	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO184	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO183	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO182	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO181	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-81. GPFMUX2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9-8	GPIO180	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO179	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO178	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO177	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
1-0	GPIO176	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.70 GPFPUD Register (Offset = 298h) [Reset = FFFFFFFFh]

GPFPUD is shown in Figure 15-73 and described in Table 15-82.

Return to the Summary Table.

GPIO F Pull Up Disable Register (GPIO160 to 191)
Disables the Pull-Up on GPIO.
0: Enables the Pull-Up.
1: Disables the Pull-Up.
Reading the register returns the current value of the register setting.
Note:
[1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low.

Figure 15-73. GPFPUD Register

31	30	29	28	27	26	25	24
GPIO191	GPIO190	GPIO189	GPIO188	GPIO187	GPIO186	GPIO185	GPIO184
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
GPIO183	GPIO182	GPIO181	GPIO180	GPIO179	GPIO178	GPIO177	GPIO176
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
GPIO175	GPIO174	GPIO173	GPIO172	GPIO171	GPIO170	GPIO169	GPIO168
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
GPIO167	GPIO166	GPIO165	GPIO164	GPIO163	GPIO162	GPIO161	GPIO160
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 15-82. GPFPUD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO191	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
30	GPIO190	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
29	GPIO189	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
28	GPIO188	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
27	GPIO187	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
26	GPIO186	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
25	GPIO185	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
24	GPIO184	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
23	GPIO183	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
22	GPIO182	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
21	GPIO181	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn

Table 15-82. GPFPUD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO180	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
19	GPIO179	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
18	GPIO178	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
17	GPIO177	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
16	GPIO176	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
15	GPIO175	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
14	GPIO174	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
13	GPIO173	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
12	GPIO172	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
11	GPIO171	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
10	GPIO170	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
9	GPIO169	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
8	GPIO168	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
7	GPIO167	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
6	GPIO166	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
5	GPIO165	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
4	GPIO164	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
3	GPIO163	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
2	GPIO162	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
1	GPIO161	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
0	GPIO160	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn

15.11.2.71 GPFINV Register (Offset = 2A0h) [Reset = 00000000h]

GPFINV is shown in Figure 15-74 and described in Table 15-83.

Return to the Summary Table.

GPIO F Input Polarity Invert Registers (GPIO160 to 191)
Selects between non-inverted and inverted GPIO input to the device.
0: selects non-inverted GPIO input
1: selects inverted GPIO input
Reading the register returns the current value of the register setting.

Figure 15-74. GPFINV Register

31	30	29	28	27	26	25	24
GPIO191	GPIO190	GPIO189	GPIO188	GPIO187	GPIO186	GPIO185	GPIO184
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO183	GPIO182	GPIO181	GPIO180	GPIO179	GPIO178	GPIO177	GPIO176
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO175	GPIO174	GPIO173	GPIO172	GPIO171	GPIO170	GPIO169	GPIO168
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO167	GPIO166	GPIO165	GPIO164	GPIO163	GPIO162	GPIO161	GPIO160
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-83. GPFINV Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO191	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
30	GPIO190	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
29	GPIO189	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
28	GPIO188	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
27	GPIO187	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
26	GPIO186	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
25	GPIO185	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
24	GPIO184	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
23	GPIO183	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
22	GPIO182	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
21	GPIO181	R/W	0h	Input inversion control for this pin Reset type: SYSRSn

Table 15-83. GPFINV Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO180	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
19	GPIO179	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
18	GPIO178	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
17	GPIO177	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
16	GPIO176	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
15	GPIO175	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
14	GPIO174	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
13	GPIO173	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
12	GPIO172	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
11	GPIO171	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
10	GPIO170	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
9	GPIO169	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
8	GPIO168	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
7	GPIO167	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
6	GPIO166	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
5	GPIO165	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
4	GPIO164	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
3	GPIO163	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
2	GPIO162	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
1	GPIO161	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
0	GPIO160	R/W	0h	Input inversion control for this pin Reset type: SYSRSn

15.11.2.72 GPFAMSEL Register (Offset = 2A8h) [Reset = FFFFFFFFh]

GPFAMSEL is shown in Figure 15-75 and described in Table 15-84.

Return to the Summary Table.

GPIO F Analog Mode Select register (GPIO160 to GPIO 191)
Selects between digital and analog functionality for GPIO pins.
0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers
1: The analog function of the pin is enabled and the pin is capable of analog functions
Reading the register returns the current value of the register setting.
Note:
[1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, t

Figure 15-75. GPFAMSEL Register

31	30	29	28	27	26	25	24
GPIO191	GPIO190	GPIO189	GPIO188	GPIO187	GPIO186	GPIO185	GPIO184
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
GPIO183	GPIO182	GPIO181	GPIO180	GPIO179	GPIO178	GPIO177	GPIO176
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
GPIO175	GPIO174	GPIO173	GPIO172	GPIO171	GPIO170	GPIO169	GPIO168
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
GPIO167	GPIO166	GPIO165	GPIO164	GPIO163	GPIO162	GPIO161	GPIO160
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 15-84. GPFAMSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO191	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
30	GPIO190	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
29	GPIO189	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
28	GPIO188	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
27	GPIO187	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
26	GPIO186	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
25	GPIO185	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
24	GPIO184	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
23	GPIO183	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
22	GPIO182	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn

Table 15-84. GPFAMSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	GPIO181	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
20	GPIO180	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
19	GPIO179	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
18	GPIO178	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
17	GPIO177	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
16	GPIO176	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
15	GPIO175	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
14	GPIO174	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
13	GPIO173	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
12	GPIO172	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
11	GPIO171	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
10	GPIO170	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
9	GPIO169	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
8	GPIO168	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
7	GPIO167	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
6	GPIO166	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
5	GPIO165	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
4	GPIO164	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
3	GPIO163	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
2	GPIO162	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
1	GPIO161	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
0	GPIO160	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn

15.11.2.73 GPFGMUX1 Register (Offset = 2C0h) [Reset = 00000000h]

GPFGMUX1 is shown in Figure 15-76 and described in Table 15-85.

Return to the Summary Table.

GPIO F Peripheral Group Mux (GPIO160 to 175)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 15-76. GPFGMUX1 Register

31	30	29	28	27	26	25	24
GPIO175		GPIO174		GPIO173		GPIO172	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
GPIO171		GPIO170		GPIO169		GPIO168	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO167		GPIO166		GPIO165		GPIO164	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO163		GPIO162		GPIO161		GPIO160	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-85. GPFGMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO175	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO174	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO173	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO172	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO171	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO170	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO169	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO168	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO167	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO166	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO165	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO164	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-85. GPFGMUX1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-6	GPIO163	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO162	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO161	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
1-0	GPIO160	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.74 GPFGMUX2 Register (Offset = 2C4h) [Reset = 00000000h]

GPFGMUX2 is shown in Figure 15-77 and described in Table 15-86.

Return to the Summary Table.

GPIO F Peripheral Group Mux (GPIO176 to 191)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 15-77. GPFGMUX2 Register

31	30	29	28	27	26	25	24
GPIO191		GPIO190		GPIO189		GPIO188	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
GPIO187		GPIO186		GPIO185		GPIO184	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO183		GPIO182		GPIO181		GPIO180	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO179		GPIO178		GPIO177		GPIO176	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-86. GPFGMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO191	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO190	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO189	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO188	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO187	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO186	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO185	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO184	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO183	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO182	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO181	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO180	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-86. GPFGMUX2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-6	GPIO179	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO178	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO177	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
1-0	GPIO176	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.75 GPFCSEL1 Register (Offset = 2D0h) [Reset = 00000000h]

GPFCSEL1 is shown in Figure 15-78 and described in Table 15-87.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-78. GPFCSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO167				GPIO166				GPIO165				GPIO164			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO163				GPIO162				GPIO161				GPIO160			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-87. GPFCSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO167	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO166	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO165	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO164	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO163	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO162	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO161	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO160	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.76 GPFCSEL2 Register (Offset = 2D4h) [Reset = 00000000h]

GPFCSEL2 is shown in Figure 15-79 and described in Table 15-88.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-79. GPFCSEL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO175				GPIO174				GPIO173				GPIO172			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO171				GPIO170				GPIO169				GPIO168			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-88. GPFCSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO175	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO174	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO173	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO172	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO171	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO170	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO169	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO168	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.77 GPFCSEL3 Register (Offset = 2D8h) [Reset = 00000000h]

GPFCSEL3 is shown in Figure 15-80 and described in Table 15-89.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-80. GPFCSEL3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO183				GPIO182				GPIO181				GPIO180			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO179				GPIO178				GPIO177				GPIO176			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-89. GPFCSEL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO183	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO182	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO181	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO180	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO179	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO178	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO177	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO176	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.78 GPFCSEL4 Register (Offset = 2DCh) [Reset = 00000000h]

GPFCSEL4 is shown in Figure 15-81 and described in Table 15-90.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-81. GPFCSEL4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO191				GPIO190				GPIO189				GPIO188			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO187				GPIO186				GPIO185				GPIO184			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-90. GPFCSEL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO191	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO190	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO189	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO188	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO187	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO186	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO185	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO184	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.79 GPFLOCK Register (Offset = 2F8h) [Reset = 00000000h]

GPFLOCK is shown in Figure 15-82 and described in Table 15-91.

Return to the Summary Table.

GPIO F Lock Configuration Register (GPIO160 to 191)
GPIO Configuration Lock for GPIO.
0: Bits in GPyMUX1, GPyMUX2, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed
1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin

Figure 15-82. GPFLOCK Register

31	30	29	28	27	26	25	24
GPIO191	GPIO190	GPIO189	GPIO188	GPIO187	GPIO186	GPIO185	GPIO184
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO183	GPIO182	GPIO181	GPIO180	GPIO179	GPIO178	GPIO177	GPIO176
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO175	GPIO174	GPIO173	GPIO172	GPIO171	GPIO170	GPIO169	GPIO168
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO167	GPIO166	GPIO165	GPIO164	GPIO163	GPIO162	GPIO161	GPIO160
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-91. GPFLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO191	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
30	GPIO190	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
29	GPIO189	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
28	GPIO188	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
27	GPIO187	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
26	GPIO186	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
25	GPIO185	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
24	GPIO184	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
23	GPIO183	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
22	GPIO182	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
21	GPIO181	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn

Table 15-91. GPFLOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO180	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
19	GPIO179	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
18	GPIO178	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
17	GPIO177	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
16	GPIO176	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
15	GPIO175	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
14	GPIO174	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
13	GPIO173	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
12	GPIO172	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
11	GPIO171	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
10	GPIO170	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
9	GPIO169	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
8	GPIO168	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
7	GPIO167	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
6	GPIO166	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
5	GPIO165	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
4	GPIO164	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
3	GPIO163	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
2	GPIO162	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
1	GPIO161	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
0	GPIO160	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn

15.11.2.80 GPFCR Register (Offset = 2FCh) [Reset = 00000000h]

GPFCR is shown in Figure 15-83 and described in Table 15-92.

Return to the Summary Table.

GPIO F Lock Commit Register (GPIO160 to 191)
GPIO Configuration Lock Commit for GPIO:
1: Locks changes to the bit in GPyLOCK register which controls the same pin
0: Bit in the GPyLOCK register which controls the same pin can be changed

Figure 15-83. GPFCR Register

31	30	29	28	27	26	25	24
GPIO191	GPIO190	GPIO189	GPIO188	GPIO187	GPIO186	GPIO185	GPIO184
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
GPIO183	GPIO182	GPIO181	GPIO180	GPIO179	GPIO178	GPIO177	GPIO176
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
GPIO175	GPIO174	GPIO173	GPIO172	GPIO171	GPIO170	GPIO169	GPIO168
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
GPIO167	GPIO166	GPIO165	GPIO164	GPIO163	GPIO162	GPIO161	GPIO160
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 15-92. GPFCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO191	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
30	GPIO190	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
29	GPIO189	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
28	GPIO188	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
27	GPIO187	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
26	GPIO186	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
25	GPIO185	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
24	GPIO184	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
23	GPIO183	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
22	GPIO182	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
21	GPIO181	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
20	GPIO180	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn

Table 15-92. GPFCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO179	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
18	GPIO178	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
17	GPIO177	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
16	GPIO176	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
15	GPIO175	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
14	GPIO174	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
13	GPIO173	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
12	GPIO172	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
11	GPIO171	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
10	GPIO170	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
9	GPIO169	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
8	GPIO168	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
7	GPIO167	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
6	GPIO166	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
5	GPIO165	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
4	GPIO164	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
3	GPIO163	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
2	GPIO162	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
1	GPIO161	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
0	GPIO160	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn

15.11.2.81 GPGCTRL Register (Offset = 300h) [Reset = 00000000h]

GPGCTRL is shown in Figure 15-84 and described in Table 15-93.

Return to the Summary Table.

GPIO G Qualification Sampling Period Control (GPIO192 to 223)

Figure 15-84. GPGCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUALPRD3								QUALPRD2								QUALPRD1								QUALPRD0							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 15-93. GPGCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	QUALPRD3	R/W	0h	Qualification sampling period for GPIO216 to GPIO223: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
23-16	QUALPRD2	R/W	0h	Qualification sampling period for GPIO208 to GPIO215: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
15-8	QUALPRD1	R/W	0h	Qualification sampling period for GPIO200 to GPIO207: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
7-0	QUALPRD0	R/W	0h	Qualification sampling period for GPIO192 to GPIO199: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn

15.11.2.82 GPGQSEL1 Register (Offset = 304h) [Reset = 00000000h]

GPGQSEL1 is shown in Figure 15-85 and described in Table 15-94.

Return to the Summary Table.

GPIO G Qualifier Select 1 Register (GPIO192 to 207)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 15-85. GPGQSEL1 Register

31	30	29	28	27	26	25	24
GPIO207		GPIO206		GPIO205		GPIO204	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
GPIO203		GPIO202		GPIO201		GPIO200	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO199		GPIO198		GPIO197		GPIO196	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO195		GPIO194		GPIO193		GPIO192	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-94. GPGQSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO207	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
29-28	GPIO206	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
27-26	GPIO205	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
25-24	GPIO204	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-94. GPGQSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-22	GPIO203	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
21-20	GPIO202	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
19-18	GPIO201	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
17-16	GPIO200	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
15-14	GPIO199	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
13-12	GPIO198	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
11-10	GPIO197	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
9-8	GPIO196	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
7-6	GPIO195	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-94. GPGQSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	GPIO194	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	GPIO193	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
1-0	GPIO192	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

15.11.2.83 GPGQSEL2 Register (Offset = 308h) [Reset = F0000000h]

GPGQSEL2 is shown in Figure 15-86 and described in Table 15-95.

Return to the Summary Table.

GPIO G Qualifier Select 2 Register (GPIO208 to 223)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 15-86. GPGQSEL2 Register

31	30	29	28	27	26	25	24
GPIO223		GPIO222		GPIO221		GPIO220	
R/W-3h		R/W-3h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
GPIO219		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED		RESERVED		GPIO213		GPIO212	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO211		GPIO210		GPIO209		GPIO208	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-95. GPGQSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO223	R/W	3h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
29-28	GPIO222	R/W	3h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
27-26	GPIO221	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
25-24	GPIO220	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-95. GPGQSEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-22	GPIO219	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	GPIO213	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
9-8	GPIO212	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
7-6	GPIO211	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
5-4	GPIO210	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	GPIO209	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
1-0	GPIO208	R/W	0h	Select input qualification type for this GPIO: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

15.11.2.84 GPGMUX1 Register (Offset = 30Ch) [Reset = 00000000h]

GPGMUX1 is shown in Figure 15-87 and described in Table 15-96.

Return to the Summary Table.

GPIO G Mux 1 Register (GPIO192 to 207)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 15-87. GPGMUX1 Register

31	30	29	28	27	26	25	24
GPIO207		GPIO206		GPIO205		GPIO204	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
GPIO203		GPIO202		GPIO201		GPIO200	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO199		GPIO198		GPIO197		GPIO196	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO195		GPIO194		GPIO193		GPIO192	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-96. GPGMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO207	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO206	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO205	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO204	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO203	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO202	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO201	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO200	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO199	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO198	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO197	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-96. GPGMUX1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9-8	GPIO196	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO195	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO194	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO193	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
1-0	GPIO192	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.85 GPGMUX2 Register (Offset = 310h) [Reset = 50000000h]

GPGMUX2 is shown in Figure 15-88 and described in Table 15-97.

Return to the Summary Table.

GPIO G Mux 2 Register (GPIO208 to 223)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 15-88. GPGMUX2 Register

31	30	29	28	27	26	25	24
GPIO223		GPIO222		GPIO221		GPIO220	
R/W-1h		R/W-1h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
GPIO219		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED		RESERVED		GPIO213		GPIO212	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO211		GPIO210		GPIO209		GPIO208	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-97. GPGMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO223	R/W	1h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO222	R/W	1h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO221	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO220	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO219	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	GPIO213	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO212	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO211	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO210	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-97. GPGMUX2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	GPIO209	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
1-0	GPIO208	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.86 GPGPUD Register (Offset = 318h) [Reset = FFFFFFFFh]

GPGPUD is shown in Figure 15-89 and described in Table 15-98.

Return to the Summary Table.

GPIO G Pull Up Disable Register (GPIO192 to 223)
Disables the Pull-Up on GPIO.
0: Enables the Pull-Up.
1: Disables the Pull-Up.
Reading the register returns the current value of the register setting.
Note:
[1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low.

Figure 15-89. GPGPUD Register

31	30	29	28	27	26	25	24
GPIO223	GPIO222	GPIO221	GPIO220	GPIO219	RESERVED	RESERVED	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	GPIO213	GPIO212	GPIO211	GPIO210	GPIO209	GPIO208
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
GPIO207	GPIO206	GPIO205	GPIO204	GPIO203	GPIO202	GPIO201	GPIO200
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
GPIO199	GPIO198	GPIO197	GPIO196	GPIO195	GPIO194	GPIO193	GPIO192
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 15-98. GPGPUD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO223	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
30	GPIO222	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
29	GPIO221	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
28	GPIO220	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
27	GPIO219	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
26	RESERVED	R/W	1h	Reserved
25	RESERVED	R/W	1h	Reserved
24	RESERVED	R/W	1h	Reserved
23	RESERVED	R/W	1h	Reserved
22	RESERVED	R/W	1h	Reserved
21	GPIO213	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
20	GPIO212	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
19	GPIO211	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn

Table 15-98. GPGPUD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	GPIO210	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
17	GPIO209	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
16	GPIO208	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
15	GPIO207	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
14	GPIO206	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
13	GPIO205	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
12	GPIO204	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
11	GPIO203	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
10	GPIO202	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
9	GPIO201	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
8	GPIO200	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
7	GPIO199	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
6	GPIO198	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
5	GPIO197	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
4	GPIO196	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
3	GPIO195	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
2	GPIO194	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
1	GPIO193	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
0	GPIO192	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn

15.11.2.87 GPGINV Register (Offset = 320h) [Reset = 00000000h]

GPGINV is shown in Figure 15-90 and described in Table 15-99.

Return to the Summary Table.

GPIO G Input Polarity Invert Registers (GPIO192 to 223)
Selects between non-inverted and inverted GPIO input to the device.
0: selects non-inverted GPIO input
1: selects inverted GPIO input
Reading the register returns the current value of the register setting.

Figure 15-90. GPGINV Register

31	30	29	28	27	26	25	24
GPIO223	GPIO222	GPIO221	GPIO220	GPIO219	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	GPIO213	GPIO212	GPIO211	GPIO210	GPIO209	GPIO208
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO207	GPIO206	GPIO205	GPIO204	GPIO203	GPIO202	GPIO201	GPIO200
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO199	GPIO198	GPIO197	GPIO196	GPIO195	GPIO194	GPIO193	GPIO192
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-99. GPGINV Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO223	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
30	GPIO222	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
29	GPIO221	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
28	GPIO220	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
27	GPIO219	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	GPIO213	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
20	GPIO212	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
19	GPIO211	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
18	GPIO210	R/W	0h	Input inversion control for this pin Reset type: SYSRSn

Table 15-99. GPGINV Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	GPIO209	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
16	GPIO208	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
15	GPIO207	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
14	GPIO206	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
13	GPIO205	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
12	GPIO204	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
11	GPIO203	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
10	GPIO202	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
9	GPIO201	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
8	GPIO200	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
7	GPIO199	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
6	GPIO198	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
5	GPIO197	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
4	GPIO196	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
3	GPIO195	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
2	GPIO194	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
1	GPIO193	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
0	GPIO192	R/W	0h	Input inversion control for this pin Reset type: SYSRSn

15.11.2.88 GPGODR Register (Offset = 324h) [Reset = 00000000h]

GPGODR is shown in Figure 15-91 and described in Table 15-100.

Return to the Summary Table.

GPIO G Open Drain Output Register (GPIO92 to 223)
Selects between normal and open-drain output for the GPIO pin.
0: Normal Output
1: Open Drain Output
Reading the register returns the current value of the register setting.
Note:
[1] In the Open Drain output mode, if the buffer is configured for output mode, a 0 value to be driven out comes out on the on the PAD while a 1 value to be driven out tri-states the buffer.

Figure 15-91. GPGODR Register

31	30	29	28	27	26	25	24
GPIO223	GPIO222	GPIO221	GPIO220	GPIO219	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-100. GPGODR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO223	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
30	GPIO222	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
29	GPIO221	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
28	GPIO220	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
27	GPIO219	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved

Table 15-100. GPGODR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	RESERVED	R/W	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

15.11.2.89 GPGAMSEL Register (Offset = 328h) [Reset = 003FFFFFh]

GPGAMSEL is shown in Figure 15-92 and described in Table 15-101.

Return to the Summary Table.

GPIO G Analog Mode Select register (GPIO192 to 223)
Selects between digital and analog functionality for GPIO pins.
0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers
1: The analog function of the pin is enabled and the pin is capable of analog functions
Reading the register returns the current value of the register setting.
Note:
[1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, t

Figure 15-92. GPGAMSEL Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	GPIO213	GPIO212	GPIO211	GPIO210	GPIO209	GPIO208
R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
GPIO207	GPIO206	GPIO205	GPIO204	GPIO203	GPIO202	GPIO201	GPIO200
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
GPIO199	GPIO198	GPIO197	GPIO196	GPIO195	GPIO194	GPIO193	GPIO192
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 15-101. GPGAMSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	GPIO213	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
20	GPIO212	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
19	GPIO211	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
18	GPIO210	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn

Table 15-101. GPGAMSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	GPIO209	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
16	GPIO208	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
15	GPIO207	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
14	GPIO206	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
13	GPIO205	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
12	GPIO204	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
11	GPIO203	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
10	GPIO202	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
9	GPIO201	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
8	GPIO200	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
7	GPIO199	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
6	GPIO198	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
5	GPIO197	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
4	GPIO196	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
3	GPIO195	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
2	GPIO194	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
1	GPIO193	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
0	GPIO192	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn

15.11.2.90 GPGGMUX1 Register (Offset = 340h) [Reset = 00000000h]

GPGGMUX1 is shown in Figure 15-93 and described in Table 15-102.

Return to the Summary Table.

GPIO G Peripheral Group Mux (GPIO192 to 207)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 15-93. GPGGMUX1 Register

31	30	29	28	27	26	25	24
GPIO207		GPIO206		GPIO205		GPIO204	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
GPIO203		GPIO202		GPIO201		GPIO200	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO199		GPIO198		GPIO197		GPIO196	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO195		GPIO194		GPIO193		GPIO192	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-102. GPGGMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO207	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO206	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO205	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO204	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO203	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO202	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO201	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO200	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO199	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO198	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO197	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO196	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-102. GPGGMUX1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-6	GPIO195	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO194	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO193	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
1-0	GPIO192	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.91 GPGGMUX2 Register (Offset = 344h) [Reset = 00000000h]

GPGGMUX2 is shown in Figure 15-94 and described in Table 15-103.

Return to the Summary Table.

GPIO G Peripheral Group Mux (GPIO208 to 223)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 15-94. GPGGMUX2 Register

31	30	29	28	27	26	25	24
GPIO223		GPIO222		GPIO221		GPIO220	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
GPIO219		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED		RESERVED		GPIO213		GPIO212	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO211		GPIO210		GPIO209		GPIO208	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-103. GPGGMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO223	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO222	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO221	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO220	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO219	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	GPIO213	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO212	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO211	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO210	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-103. GPGGMUX2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	GPIO209	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
1-0	GPIO208	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.92 GPGCSEL1 Register (Offset = 350h) [Reset = 00000000h]

GPGCSEL1 is shown in Figure 15-95 and described in Table 15-104.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-95. GPGCSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO199				GPIO198				GPIO197				GPIO196			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO195				GPIO194				GPIO193				GPIO192			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-104. GPGCSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO199	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO198	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO197	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO196	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO195	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO194	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO193	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO192	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.93 GPGCSEL2 Register (Offset = 354h) [Reset = 00000000h]

GPGCSEL2 is shown in Figure 15-96 and described in Table 15-105.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-96. GPGCSEL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO207				GPIO206				GPIO205				GPIO204			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO203				GPIO202				GPIO201				GPIO200			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-105. GPGCSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO207	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO206	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO205	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO204	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO203	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO202	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO201	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO200	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.94 GPGCSEL3 Register (Offset = 358h) [Reset = 00000000h]

GPGCSEL3 is shown in Figure 15-97 and described in Table 15-106.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-97. GPGCSEL3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				RESERVED				GPIO213				GPIO212			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO211				GPIO210				GPIO209				GPIO208			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-106. GPGCSEL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	Reserved
27-24	RESERVED	R/W	0h	Reserved
23-20	GPIO213	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO212	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO211	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO210	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO209	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO208	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.95 GPGCSEL4 Register (Offset = 35Ch) [Reset = 00000000h]

GPGCSEL4 is shown in Figure 15-98 and described in Table 15-107.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-98. GPGCSEL4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO223				GPIO222				GPIO221				GPIO220			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO219				RESERVED				RESERVED				RESERVED			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-107. GPGCSEL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO223	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO222	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO221	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO220	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO219	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	RESERVED	R/W	0h	Reserved
7-4	RESERVED	R/W	0h	Reserved
3-0	RESERVED	R/W	0h	Reserved

15.11.2.96 GPGLOCK Register (Offset = 378h) [Reset = 00000000h]

GPGLOCK is shown in Figure 15-99 and described in Table 15-108.

Return to the Summary Table.

GPIO G Lock Configuration Register (GPIO192 to 223)
GPIO Configuration Lock for GPIO.
0: Bits in GPyMUX1, GPyMUX2, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed
1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin

Figure 15-99. GPGLOCK Register

31	30	29	28	27	26	25	24
GPIO223	GPIO222	GPIO221	GPIO220	GPIO219	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	GPIO213	GPIO212	GPIO211	GPIO210	GPIO209	GPIO208
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO207	GPIO206	GPIO205	GPIO204	GPIO203	GPIO202	GPIO201	GPIO200
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO199	GPIO198	GPIO197	GPIO196	GPIO195	GPIO194	GPIO193	GPIO192
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-108. GPGLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO223	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
30	GPIO222	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
29	GPIO221	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
28	GPIO220	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
27	GPIO219	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	GPIO213	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
20	GPIO212	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
19	GPIO211	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn

Table 15-108. GPGLOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	GPIO210	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
17	GPIO209	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
16	GPIO208	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
15	GPIO207	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
14	GPIO206	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
13	GPIO205	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
12	GPIO204	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
11	GPIO203	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
10	GPIO202	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
9	GPIO201	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
8	GPIO200	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
7	GPIO199	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
6	GPIO198	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
5	GPIO197	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
4	GPIO196	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
3	GPIO195	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
2	GPIO194	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
1	GPIO193	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
0	GPIO192	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn

15.11.2.97 GPGCR Register (Offset = 37Ch) [Reset = 00000000h]

GPGCR is shown in Figure 15-100 and described in Table 15-109.

Return to the Summary Table.

GPIO G Lock Commit Register (GPIO192 to 223)
GPIO Configuration Lock Commit for GPIO:
1: Locks changes to the bit in GPyLOCK register which controls the same pin
0: Bit in the GPyLOCK register which controls the same pin can be changed

Figure 15-100. GPGCR Register

31	30	29	28	27	26	25	24
GPIO223	GPIO222	GPIO221	GPIO220	GPIO219	RESERVED	RESERVED	RESERVED
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	GPIO213	GPIO212	GPIO211	GPIO210	GPIO209	GPIO208
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
GPIO207	GPIO206	GPIO205	GPIO204	GPIO203	GPIO202	GPIO201	GPIO200
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
GPIO199	GPIO198	GPIO197	GPIO196	GPIO195	GPIO194	GPIO193	GPIO192
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 15-109. GPGCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO223	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
30	GPIO222	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
29	GPIO221	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
28	GPIO220	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
27	GPIO219	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
26	RESERVED	R/WSonce	0h	Reserved
25	RESERVED	R/WSonce	0h	Reserved
24	RESERVED	R/WSonce	0h	Reserved
23	RESERVED	R/WSonce	0h	Reserved
22	RESERVED	R/WSonce	0h	Reserved
21	GPIO213	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
20	GPIO212	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
19	GPIO211	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
18	GPIO210	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn

Table 15-109. GPGCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	GPIO209	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
16	GPIO208	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
15	GPIO207	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
14	GPIO206	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
13	GPIO205	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
12	GPIO204	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
11	GPIO203	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
10	GPIO202	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
9	GPIO201	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
8	GPIO200	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
7	GPIO199	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
6	GPIO198	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
5	GPIO197	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
4	GPIO196	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
3	GPIO195	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
2	GPIO194	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
1	GPIO193	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
0	GPIO192	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn

15.11.2.98 GPHCTRL Register (Offset = 380h) [Reset = 00000000h]

GPHCTRL is shown in Figure 15-101 and described in Table 15-110.

Return to the Summary Table.

GPIO H Qualification Sampling Period Control (GPIO224 to 255)

Figure 15-101. GPHCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUALPRD3								QUALPRD2								QUALPRD1								QUALPRD0							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 15-110. GPHCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	QUALPRD3	R/W	0h	0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/513 Reset type: SYSRSn
23-16	QUALPRD2	R/W	0h	0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/512 Reset type: SYSRSn
15-8	QUALPRD1	R/W	0h	0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/511 Reset type: SYSRSn
7-0	QUALPRD0	R/W	0h	0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn

15.11.2.99 GPHQSEL1 Register (Offset = 384h) [Reset = 00000000h]

GPHQSEL1 is shown in Figure 15-102 and described in Table 15-111.

Return to the Summary Table.

GPIO H Qualifier Select 1 Register (GPIO224 to 239)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 15-102. GPHQSEL1 Register

31	30	29	28	27	26	25	24
GPIO239		GPIO238		GPIO237		GPIO236	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
GPIO235		GPIO234		GPIO233		GPIO232	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO231		GPIO230		GPIO229		GPIO228	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO227		GPIO226		GPIO225		GPIO224	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-111. GPHQSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO239	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
29-28	GPIO238	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
27-26	GPIO237	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
25-24	GPIO236	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
23-22	GPIO235	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-111. GPHQSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21-20	GPIO234	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
19-18	GPIO233	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
17-16	GPIO232	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
15-14	GPIO231	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
13-12	GPIO230	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
11-10	GPIO229	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
9-8	GPIO228	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
7-6	GPIO227	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
5-4	GPIO226	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	GPIO225	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
1-0	GPIO224	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

15.11.2.100 GPHQSEL2 Register (Offset = 388h) [Reset = 00000000h]

GPHQSEL2 is shown in Figure 15-103 and described in Table 15-112.

Return to the Summary Table.

GPIO H Qualifier Select 2 Register (GPIO240 to 255)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 15-103. GPHQSEL2 Register

31	30	29	28	27	26	25	24
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		RESERVED		GPIO249		GPIO248	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO247		GPIO246		GPIO245		GPIO244	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO243		GPIO242		GPIO241		GPIO240	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-112. GPHQSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	GPIO249	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
17-16	GPIO248	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
15-14	GPIO247	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 15-112. GPHQSEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-12	GPIO246	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
11-10	GPIO245	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
9-8	GPIO244	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
7-6	GPIO243	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
5-4	GPIO242	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	GPIO241	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
1-0	GPIO240	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

15.11.2.101 GPHMUX1 Register (Offset = 38Ch) [Reset = 00000000h]

GPHMUX1 is shown in Figure 15-104 and described in Table 15-113.

Return to the Summary Table.

GPIO H Mux 1 Register (GPIO224 to 239)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 15-104. GPHMUX1 Register

31	30	29	28	27	26	25	24
GPIO239		GPIO238		GPIO237		GPIO236	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
GPIO235		GPIO234		GPIO233		GPIO232	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO231		GPIO230		GPIO229		GPIO228	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO227		GPIO226		GPIO225		GPIO224	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-113. GPHMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO239	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO238	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO237	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO236	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO235	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO234	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO233	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO232	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO231	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO230	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO229	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-113. GPHMUX1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9-8	GPIO228	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO227	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO226	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO225	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
1-0	GPIO224	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.102 GPHMUX2 Register (Offset = 390h) [Reset = 00000000h]

GPHMUX2 is shown in Figure 15-105 and described in Table 15-114.

Return to the Summary Table.

GPIO H Mux 2 Register (GPIO240 to 255)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 15-105. GPHMUX2 Register

31	30	29	28	27	26	25	24
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		RESERVED		GPIO249		GPIO248	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO247		GPIO246		GPIO245		GPIO244	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO243		GPIO242		GPIO241		GPIO240	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-114. GPHMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	GPIO249	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO248	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO247	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO246	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO245	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO244	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO243	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO242	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-114. GPHMUX2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	GPIO241	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
1-0	GPIO240	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.103 GPHPUD Register (Offset = 398h) [Reset = FFFFFFFFh]

GPHPUD is shown in Figure 15-106 and described in Table 15-115.

Return to the Summary Table.

GPIO H Pull Up Disable Register (GPIO224 to 255)
Disables the Pull-Up on GPIO.
0: Enables the Pull-Up.
1: Disables the Pull-Up.
Reading the register returns the current value of the register setting.
Note:
[1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low.

Figure 15-106. GPHPUD Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO249	GPIO248
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
GPIO247	GPIO246	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	GPIO240
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	GPIO236	GPIO235	GPIO234	GPIO233	GPIO232
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	GPIO229	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 15-115. GPHPUD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	1h	Reserved
30	RESERVED	R/W	1h	Reserved
29	RESERVED	R/W	1h	Reserved
28	RESERVED	R/W	1h	Reserved
27	RESERVED	R/W	1h	Reserved
26	RESERVED	R/W	1h	Reserved
25	GPIO249	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
24	GPIO248	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn

Table 15-115. GPHPUD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	GPIO247	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
22	GPIO246	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
21	GPIO245	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
20	GPIO244	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
19	GPIO243	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
18	GPIO242	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn

Table 15-115. GPHPUD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	GPIO241	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
16	GPIO240	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
15	GPIO239	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
14	GPIO238	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
13	GPIO237	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
12	GPIO236	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn

Table 15-115. GPHPUD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	GPIO235	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
10	GPIO234	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
9	GPIO233	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
8	GPIO232	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
7	GPIO231	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
6	GPIO230	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn

Table 15-115. GPHPUD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	GPIO229	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
4	GPIO228	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
3	GPIO227	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
2	GPIO226	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
1	GPIO225	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
0	GPIO224	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn

15.11.2.104 GPHINV Register (Offset = 3A0h) [Reset = 00000000h]

GPHINV is shown in Figure 15-107 and described in Table 15-116.

Return to the Summary Table.

GPIO H Input Polarity Invert Registers (GPIO224 to 255)
Selects between non-inverted and inverted GPIO input to the device.
0: selects non-inverted GPIO input
1: selects inverted GPIO input
Reading the register returns the current value of the register setting.

Figure 15-107. GPHINV Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO249	GPIO248
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO247	GPIO246	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	GPIO240
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	GPIO236	GPIO235	GPIO234	GPIO233	GPIO232
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	GPIO229	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-116. GPHINV Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	GPIO249	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
24	GPIO248	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
23	GPIO247	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn

Table 15-116. GPHINV Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	GPIO246	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
21	GPIO245	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
20	GPIO244	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
19	GPIO243	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
18	GPIO242	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
17	GPIO241	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
16	GPIO240	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
15	GPIO239	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
14	GPIO238	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn

Table 15-116. GPHINV Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	GPIO237	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
12	GPIO236	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
11	GPIO235	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
10	GPIO234	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
9	GPIO233	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
8	GPIO232	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
7	GPIO231	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
6	GPIO230	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
5	GPIO229	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn

Table 15-116. GPHINV Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	GPIO228	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
3	GPIO227	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
2	GPIO226	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
1	GPIO225	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
0	GPIO224	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn

15.11.2.105 GPHODR Register (Offset = 3A4h) [Reset = 00000000h]

GPHODR is shown in Figure 15-108 and described in Table 15-117.

Return to the Summary Table.

GPIO H Open Drain Output Register (GPIO224 to GPIO255)
Selects between normal and open-drain output for the GPIO pin.
0: Normal Output
1: Open Drain Output
Reading the register returns the current value of the register setting.
Note:
[1] In the Open Drain output mode, if the buffer is configured for output mode, a 0 value to be driven out comes out on the on the PAD while a 1 value to be driven out tri-states the buffer.

Figure 15-108. GPHODR Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO249	GPIO248
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO247	GPIO246	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	GPIO240
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	GPIO236	GPIO235	GPIO234	GPIO233	GPIO232
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	GPIO229	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-117. GPHODR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	GPIO249	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
24	GPIO248	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
23	GPIO247	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
22	GPIO246	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
21	GPIO245	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
20	GPIO244	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
19	GPIO243	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn

Table 15-117. GPHODR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	GPIO242	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
17	GPIO241	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
16	GPIO240	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
15	GPIO239	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
14	GPIO238	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
13	GPIO237	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
12	GPIO236	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
11	GPIO235	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
10	GPIO234	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
9	GPIO233	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
8	GPIO232	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
7	GPIO231	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
6	GPIO230	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
5	GPIO229	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
4	GPIO228	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
3	GPIO227	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
2	GPIO226	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
1	GPIO225	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn
0	GPIO224	R/W	0h	Outpout Open-Drain control for this pin Reset type: SYSRSn

15.11.2.106 GPHAMSEL Register (Offset = 3A8h) [Reset = 03FFFFFFh]

GPHAMSEL is shown in Figure 15-109 and described in Table 15-118.

Return to the Summary Table.

GPIO H Analog Mode Select register (GPIO224 to GPIO255)
Selects between digital and analog functionality for GPIO pins.
0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers
1: The analog function of the pin is enabled and the pin is capable of analog functions
Reading the register returns the current value of the register setting.
Note:
[1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, t

Figure 15-109. GPHAMSEL Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO249	GPIO248
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
GPIO247	GPIO246	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	GPIO240
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	GPIO236	GPIO235	GPIO234	GPIO233	GPIO232
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	GPIO229	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 15-118. GPHAMSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	GPIO249	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn

Table 15-118. GPHAMSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	GPIO248	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
23	GPIO247	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
22	GPIO246	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
21	GPIO245	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
20	GPIO244	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn

Table 15-118. GPHAMSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO243	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
18	GPIO242	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
17	GPIO241	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
16	GPIO240	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
15	GPIO239	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn

Table 15-118. GPHAMSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	GPIO238	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
13	GPIO237	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
12	GPIO236	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
11	GPIO235	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
10	GPIO234	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn

Table 15-118. GPHAMSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	GPIO233	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
8	GPIO232	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
7	GPIO231	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
6	GPIO230	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
5	GPIO229	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn

Table 15-118. GPHAMSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	GPIO228	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
3	GPIO227	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
2	GPIO226	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
1	GPIO225	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
0	GPIO224	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn

15.11.2.107 GPHGMUX1 Register (Offset = 3C0h) [Reset = 00000000h]

GPHGMUX1 is shown in Figure 15-110 and described in Table 15-119.

Return to the Summary Table.

GPIO H Peripheral Group Mux (GPIO224 to 239)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 15-110. GPHGMUX1 Register

31	30	29	28	27	26	25	24
GPIO239		GPIO238		GPIO237		GPIO236	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
GPIO235		GPIO234		GPIO233		GPIO232	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO231		GPIO230		GPIO229		GPIO228	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO227		GPIO226		GPIO225		GPIO224	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-119. GPHGMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO239	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
29-28	GPIO238	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO237	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO236	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO235	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO234	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO233	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO232	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO231	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO230	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO229	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO228	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-119. GPHGMUX1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-6	GPIO227	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO226	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO225	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
1-0	GPIO224	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.108 GPHGMUX2 Register (Offset = 3C4h) [Reset = 00000000h]

GPHGMUX2 is shown in Figure 15-111 and described in Table 15-120.

Return to the Summary Table.

GPIO H Peripheral Group Mux (GPIO240 to 255)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 15-111. GPHGMUX2 Register

31	30	29	28	27	26	25	24
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		RESERVED		GPIO249		GPIO248	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO247		GPIO246		GPIO245		GPIO244	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO243		GPIO242		GPIO241		GPIO240	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-120. GPHGMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	GPIO249	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO248	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO247	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO246	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO245	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO244	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO243	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO242	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO241	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

Table 15-120. GPHGMUX2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPIO240	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

15.11.2.109 GPHCSEL1 Register (Offset = 3D0h) [Reset = 00000000h]

GPHCSEL1 is shown in Figure 15-112 and described in Table 15-121.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-112. GPHCSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO231				GPIO230				GPIO229				GPIO228			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO227				GPIO226				GPIO225				GPIO224			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-121. GPHCSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO231	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO230	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO229	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO228	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO227	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO226	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO225	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO224	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.110 GPHCSEL2 Register (Offset = 3D4h) [Reset = 00000000h]

GPHCSEL2 is shown in Figure 15-113 and described in Table 15-122.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-113. GPHCSEL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO239				GPIO238				GPIO237				GPIO236			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO235				GPIO234				GPIO233				GPIO232			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-122. GPHCSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO239	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO238	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO237	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO236	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO235	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO234	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO233	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO232	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.111 GPHCSEL3 Register (Offset = 3D8h) [Reset = 00000000h]

GPHCSEL3 is shown in Figure 15-114 and described in Table 15-123.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-114. GPHCSEL3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO247				GPIO246				GPIO245				GPIO244			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO243				GPIO242				GPIO241				GPIO240			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-123. GPHCSEL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	GPIO247	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
27-24	GPIO246	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
23-20	GPIO245	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
19-16	GPIO244	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
15-12	GPIO243	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
11-8	GPIO242	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
7-4	GPIO241	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO240	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.112 GPHCSEL4 Register (Offset = 3DCh) [Reset = 00000000h]

GPHCSEL4 is shown in Figure 15-115 and described in Table 15-124.

Return to the Summary Table.

Selects which controller's GPIODAT/SET/CLEAR/TOGGLE/DIR registers control this GPIO pin
0000: CPU1 selected
0001: CPU2 selected
0010: CPU3 selected
0011: CPU4 selected (Reserved)
0100: CPU5 selected (Reserved)
0101: CPU6 selected (Reserved)
1xxx: (Reserved)

Figure 15-115. GPHCSEL4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				RESERVED				RESERVED				RESERVED			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				RESERVED				GPIO249				GPIO248			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 15-124. GPHCSEL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	Reserved
27-24	RESERVED	R/W	0h	Reserved
23-20	RESERVED	R/W	0h	Reserved
19-16	RESERVED	R/W	0h	Reserved
15-12	RESERVED	R/W	0h	Reserved
11-8	RESERVED	R/W	0h	Reserved
7-4	GPIO249	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn
3-0	GPIO248	R/W	0h	Selects which controller's GPIODAT/SET/CLEAR/TOGGLE registers control this GPIO pin Reset type: SYSRSn

15.11.2.113 GPHLOCK Register (Offset = 3F8h) [Reset = 00000000h]

GPHLOCK is shown in Figure 15-116 and described in Table 15-125.

Return to the Summary Table.

GPIO H Lock Configuration Register (GPIO224 to 255)
GPIO Configuration Lock for GPIO.
0: Bits in GPyMUX1, GPyMUX2, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed
1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin

Figure 15-116. GPHLOCK Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO249	GPIO248
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO247	GPIO246	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	GPIO240
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	GPIO236	GPIO235	GPIO234	GPIO233	GPIO232
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	GPIO229	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-125. GPHLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	GPIO249	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
24	GPIO248	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
23	GPIO247	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
22	GPIO246	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
21	GPIO245	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
20	GPIO244	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
19	GPIO243	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
18	GPIO242	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn

Table 15-125. GPHLOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	GPIO241	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
16	GPIO240	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
15	GPIO239	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
14	GPIO238	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
13	GPIO237	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
12	GPIO236	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
11	GPIO235	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
10	GPIO234	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
9	GPIO233	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
8	GPIO232	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
7	GPIO231	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
6	GPIO230	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
5	GPIO229	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
4	GPIO228	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
3	GPIO227	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
2	GPIO226	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
1	GPIO225	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
0	GPIO224	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn

15.11.2.114 GPHCR Register (Offset = 3FCh) [Reset = 00000000h]

GPHCR is shown in Figure 15-117 and described in Table 15-126.

Return to the Summary Table.

GPIO H Lock Commit Register (GPIO224 to 255)
GPIO Configuration Lock Commit for GPIO:
1: Locks changes to the bit in GPyLOCK register which controls the same pin
0: Bit in the GPyLOCK register which controls the same pin can be changed

Figure 15-117. GPHCR Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO249	GPIO248
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
GPIO247	GPIO246	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	GPIO240
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	GPIO236	GPIO235	GPIO234	GPIO233	GPIO232
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	GPIO229	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 15-126. GPHCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/WSonce	0h	Reserved
30	RESERVED	R/WSonce	0h	Reserved
29	RESERVED	R/WSonce	0h	Reserved
28	RESERVED	R/WSonce	0h	Reserved
27	RESERVED	R/WSonce	0h	Reserved
26	RESERVED	R/WSonce	0h	Reserved
25	GPIO249	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
24	GPIO248	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
23	GPIO247	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
22	GPIO246	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn

Table 15-126. GPHCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	GPIO245	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
20	GPIO244	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
19	GPIO243	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
18	GPIO242	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
17	GPIO241	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
16	GPIO240	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
15	GPIO239	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
14	GPIO238	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
13	GPIO237	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
12	GPIO236	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
11	GPIO235	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn

Table 15-126. GPHCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	GPIO234	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
9	GPIO233	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
8	GPIO232	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
7	GPIO231	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
6	GPIO230	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
5	GPIO229	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
4	GPIO228	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
3	GPIO227	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
2	GPIO226	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
1	GPIO225	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
0	GPIO224	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn

15.11.3 GPIO_DATA_REGS Registers

Table 15-127 lists the memory-mapped registers for the GPIO_DATA_REGS registers. All register offset addresses not listed in Table 15-127 should be considered as reserved locations and the register contents should not be modified.

Table 15-127. GPIO_DATA_REGS Registers

Offset	Acronym	Register Name	Protection
0h	GPADAT	GPIO A Data Register (GPIO0 to 31)	
4h	GPASET	GPIO A Data Set Register (GPIO0 to 31)	
8h	GPACLEAR	GPIO A Data Clear Register (GPIO0 to 31)	
Ch	GPATOGGLE	GPIO A Data Toggle Register (GPIO0 to 31)	
10h	GPADIR	GPIO A Direction Register (GPIO0 to 31)	
14h	GPBDAT	GPIO B Data Register (GPIO32 to 63)	
18h	GPBSET	GPIO B Data Set Register (GPIO32 to 63)	
1Ch	GPBCLEAR	GPIO B Data Clear Register (GPIO32 to 63)	
20h	GPBTOGGLE	GPIO B Data Toggle Register (GPIO32 to 63)	
24h	GPBDIR	GPIO B Direction Register (GPIO0 to 31)	
28h	GPCDAT	GPIO C Data Register (GPIO64 to 95)	
2Ch	GPCSET	GPIO C Data Set Register (GPIO64 to 95)	
30h	GPCCLEAR	GPIO C Data Clear Register (GPIO64 to 95)	
34h	GPCTOGGLE	GPIO C Data Toggle Register (GPIO64 to 95)	
38h	GPCDIR	GPIO C Direction Register (GPIO0 to 31)	
3Ch	GPDDAT	GPIO D Data Register (GPIO96 to 127)	
40h	GPDSET	GPIO D Data Set Register (GPIO96 to 127)	
44h	GPDCLEAR	GPIO D Data Clear Register (GPIO96 to 127)	
48h	GPDTOGGLE	GPIO D Data Toggle Register (GPIO96 to 127)	
4Ch	GPDDIR	GPIO D Direction Register (GPIO0 to 31)	
64h	GPFDAT	GPIO F Data Register (GPIO160 to 191)	
78h	GPGDAT	GPIO G Data Register (GPIO192 to 223)	
7Ch	GPGSET	GPIO G Data Set Register (GPIO192 to 223)	
80h	GPGCLEAR	GPIO G Data Clear Register (GPIO192 to 223)	
84h	GPGTOGGLE	GPIO G Data Toggle Register (GPIO192 to 223)	
88h	GPGDIR	GPIO G Direction Register (GPIO0 to 31)	
8Ch	GPHDAT	GPIO H Data Register (GPIO224 to 255)	
90h	GPHSET	GPIO H Data Set Register (GPIO224 to 255)	
94h	GPHCLEAR	GPIO H Data Clear Register (GPIO224 to 255)	
98h	GPHTOGGLE	GPIO H Data Toggle Register (GPIO224 to 255)	
9Ch	GPHDIR	GPIO H Direction Register (GPIO0 to 31)	

Complex bit access types are encoded to fit into small table cells. Table 15-128 shows the codes that are used for access types in this section.

Table 15-128. GPIO_DATA_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		

Table 15-128. GPIO_DATA_REGS Access Type Codes (continued)

Access Type	Code	Description
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

15.11.3.1 GPADAT Register (Offset = 0h) [Reset = 00000000h]

GPADAT is shown in Figure 15-118 and described in Table 15-129.

Return to the Summary Table.

GPIO A Data Register (GPIO0 to 31)
Reading this register indicates the current status of the GPIO pin, irrespective of which mode the pin is in. Writing to this register will set the GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero.
DESIGNER NOTE:
[1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register.

Figure 15-118. GPADAT Register

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-129. GPADAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO31	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
30	GPIO30	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
29	GPIO29	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
28	GPIO28	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
27	GPIO27	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
26	GPIO26	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
25	GPIO25	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
24	GPIO24	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
23	GPIO23	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
22	GPIO22	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn

Table 15-129. GPADAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	GPIO21	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
20	GPIO20	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
19	GPIO19	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
18	GPIO18	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
17	GPIO17	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
16	GPIO16	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
15	GPIO15	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
14	GPIO14	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
13	GPIO13	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
12	GPIO12	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
11	GPIO11	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
10	GPIO10	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
9	GPIO9	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
8	GPIO8	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
7	GPIO7	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
6	GPIO6	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
5	GPIO5	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
4	GPIO4	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
3	GPIO3	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
2	GPIO2	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
1	GPIO1	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
0	GPIO0	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn

15.11.3.2 GPASET Register (Offset = 4h) [Reset = 00000000h]

GPASET is shown in Figure 15-119 and described in Table 15-130.

Return to the Summary Table.

GPIO A Data Set Register (GPIO0 to 31)
Writing a 1 will force GPIO output data latch to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-119. GPASET Register

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-130. GPASET Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO31	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
30	GPIO30	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
29	GPIO29	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
28	GPIO28	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
27	GPIO27	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
26	GPIO26	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
25	GPIO25	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
24	GPIO24	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
23	GPIO23	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
22	GPIO22	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
21	GPIO21	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
20	GPIO20	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn

Table 15-130. GPASET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO19	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
18	GPIO18	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
17	GPIO17	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
16	GPIO16	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
15	GPIO15	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
14	GPIO14	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
13	GPIO13	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
12	GPIO12	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
11	GPIO11	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
10	GPIO10	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
9	GPIO9	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
8	GPIO8	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
7	GPIO7	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
6	GPIO6	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
5	GPIO5	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
4	GPIO4	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
3	GPIO3	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
2	GPIO2	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
1	GPIO1	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
0	GPIO0	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn

15.11.3.3 GPACLEAR Register (Offset = 8h) [Reset = 00000000h]

GPACLEAR is shown in Figure 15-120 and described in Table 15-131.

Return to the Summary Table.

GPIO A Data Clear Register (GPIO0 to 31)
Writing a 1 will force GPIO0 output data latch to 0.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-120. GPACLEAR Register

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-131. GPACLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO31	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
30	GPIO30	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
29	GPIO29	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
28	GPIO28	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
27	GPIO27	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
26	GPIO26	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
25	GPIO25	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
24	GPIO24	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
23	GPIO23	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
22	GPIO22	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
21	GPIO21	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
20	GPIO20	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn

Table 15-131. GPACLEAR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO19	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
18	GPIO18	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
17	GPIO17	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
16	GPIO16	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
15	GPIO15	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
14	GPIO14	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
13	GPIO13	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
12	GPIO12	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
11	GPIO11	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
10	GPIO10	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
9	GPIO9	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
8	GPIO8	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
7	GPIO7	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
6	GPIO6	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
5	GPIO5	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
4	GPIO4	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
3	GPIO3	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
2	GPIO2	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
1	GPIO1	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
0	GPIO0	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn

15.11.3.4 GPATOGGLE Register (Offset = Ch) [Reset = 00000000h]

GPATOGGLE is shown in Figure 15-121 and described in Table 15-132.

Return to the Summary Table.

GPIO A Data Toggle Register (GPIO0 to 31)
Writing a 1 will toggle GPIO0 output data latch 1 to 0 or 0 to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-121. GPATOGGLE Register

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-132. GPATOGGLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO31	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
30	GPIO30	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
29	GPIO29	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
28	GPIO28	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
27	GPIO27	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
26	GPIO26	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
25	GPIO25	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
24	GPIO24	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
23	GPIO23	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
22	GPIO22	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
21	GPIO21	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
20	GPIO20	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn

Table 15-132. GPATOGGLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO19	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
18	GPIO18	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
17	GPIO17	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
16	GPIO16	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
15	GPIO15	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
14	GPIO14	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
13	GPIO13	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
12	GPIO12	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
11	GPIO11	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
10	GPIO10	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
9	GPIO9	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
8	GPIO8	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
7	GPIO7	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
6	GPIO6	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
5	GPIO5	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
4	GPIO4	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
3	GPIO3	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
2	GPIO2	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
1	GPIO1	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
0	GPIO0	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn

15.11.3.5 GPADIR Register (Offset = 10h) [Reset = 00000000h]

GPADIR is shown in Figure 15-122 and described in Table 15-133.

Return to the Summary Table.

GPIO A Direction Register (GPIO0 to 31)
Controls direction of GPIO pins when the specified pin is configured in GPIO mode.
0: Configures pin as input.
1: Configures pin as output.
Reading the register returns the current value of the register setting.

Figure 15-122. GPADIR Register

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-133. GPADIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO31	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
30	GPIO30	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
29	GPIO29	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
28	GPIO28	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
27	GPIO27	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
26	GPIO26	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
25	GPIO25	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
24	GPIO24	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
23	GPIO23	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
22	GPIO22	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
21	GPIO21	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn

Table 15-133. GPADIR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO20	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
19	GPIO19	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
18	GPIO18	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
17	GPIO17	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
16	GPIO16	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
15	GPIO15	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
14	GPIO14	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
13	GPIO13	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
12	GPIO12	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
11	GPIO11	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
10	GPIO10	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
9	GPIO9	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
8	GPIO8	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
7	GPIO7	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
6	GPIO6	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
5	GPIO5	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
4	GPIO4	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
3	GPIO3	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
2	GPIO2	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
1	GPIO1	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
0	GPIO0	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn

15.11.3.6 GPBDAT Register (Offset = 14h) [Reset = 00000000h]

GPBDAT is shown in Figure 15-123 and described in Table 15-134.

Return to the Summary Table.

GPIO B Data Register (GPIO32 to 63)
Reading this register indicates the current status of the GPIO pin, irrespective of which mode the pin is in. Writing to this register will set the GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero.
DESIGNER NOTE:
[1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register.

Figure 15-123. GPBDAT Register

31	30	29	28	27	26	25	24
GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-134. GPBDAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO63	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
30	GPIO62	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
29	GPIO61	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
28	GPIO60	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
27	GPIO59	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
26	GPIO58	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
25	GPIO57	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
24	GPIO56	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
23	GPIO55	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
22	GPIO54	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn

Table 15-134. GPBDAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	GPIO53	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
20	GPIO52	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
19	GPIO51	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
18	GPIO50	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
17	GPIO49	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
16	GPIO48	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
15	GPIO47	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
14	GPIO46	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
13	GPIO45	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
12	GPIO44	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
11	GPIO43	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
10	GPIO42	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
9	GPIO41	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
8	GPIO40	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
7	GPIO39	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
6	GPIO38	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
5	GPIO37	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
4	GPIO36	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
3	GPIO35	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
2	GPIO34	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
1	GPIO33	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
0	GPIO32	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn

15.11.3.7 GPBSET Register (Offset = 18h) [Reset = 00000000h]

GPBSET is shown in Figure 15-124 and described in Table 15-135.

Return to the Summary Table.

GPIO B Data Set Register (GPIO32 to 63)
Writing a 1 will force GPIO output data latch to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-124. GPBSET Register

31	30	29	28	27	26	25	24
GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-135. GPBSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO63	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
30	GPIO62	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
29	GPIO61	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
28	GPIO60	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
27	GPIO59	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
26	GPIO58	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
25	GPIO57	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
24	GPIO56	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
23	GPIO55	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
22	GPIO54	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
21	GPIO53	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
20	GPIO52	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn

Table 15-135. GPBSET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO51	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
18	GPIO50	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
17	GPIO49	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
16	GPIO48	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
15	GPIO47	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
14	GPIO46	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
13	GPIO45	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
12	GPIO44	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
11	GPIO43	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
10	GPIO42	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
9	GPIO41	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
8	GPIO40	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
7	GPIO39	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
6	GPIO38	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
5	GPIO37	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
4	GPIO36	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
3	GPIO35	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
2	GPIO34	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
1	GPIO33	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
0	GPIO32	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn

15.11.3.8 GPBCLEAR Register (Offset = 1Ch) [Reset = 00000000h]

GPBCLEAR is shown in Figure 15-125 and described in Table 15-136.

Return to the Summary Table.

GPIO B Data Clear Register (GPIO32 to 63)
Writing a 1 will force GPIO0 output data latch to 0.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-125. GPBCLEAR Register

31	30	29	28	27	26	25	24
GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-136. GPBCLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO63	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
30	GPIO62	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
29	GPIO61	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
28	GPIO60	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
27	GPIO59	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
26	GPIO58	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
25	GPIO57	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
24	GPIO56	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
23	GPIO55	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
22	GPIO54	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
21	GPIO53	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
20	GPIO52	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn

Table 15-136. GPBCLEAR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO51	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
18	GPIO50	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
17	GPIO49	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
16	GPIO48	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
15	GPIO47	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
14	GPIO46	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
13	GPIO45	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
12	GPIO44	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
11	GPIO43	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
10	GPIO42	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
9	GPIO41	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
8	GPIO40	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
7	GPIO39	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
6	GPIO38	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
5	GPIO37	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
4	GPIO36	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
3	GPIO35	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
2	GPIO34	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
1	GPIO33	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
0	GPIO32	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn

15.11.3.9 GPBTOGGLE Register (Offset = 20h) [Reset = 00000000h]

GPBTOGGLE is shown in Figure 15-126 and described in Table 15-137.

Return to the Summary Table.

GPIO B Data Toggle Register (GPIO32 to 63)
Writing a 1 will toggle GPIO0 output data latch 1 to 0 or 0 to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-126. GPBTOGGLE Register

31	30	29	28	27	26	25	24
GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-137. GPBTOGGLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO63	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
30	GPIO62	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
29	GPIO61	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
28	GPIO60	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
27	GPIO59	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
26	GPIO58	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
25	GPIO57	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
24	GPIO56	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
23	GPIO55	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
22	GPIO54	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
21	GPIO53	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
20	GPIO52	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn

Table 15-137. GPBTOGGLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO51	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
18	GPIO50	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
17	GPIO49	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
16	GPIO48	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
15	GPIO47	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
14	GPIO46	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
13	GPIO45	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
12	GPIO44	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
11	GPIO43	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
10	GPIO42	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
9	GPIO41	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
8	GPIO40	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
7	GPIO39	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
6	GPIO38	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
5	GPIO37	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
4	GPIO36	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
3	GPIO35	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
2	GPIO34	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
1	GPIO33	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
0	GPIO32	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn

15.11.3.10 GPBDIR Register (Offset = 24h) [Reset = 00000000h]

GPBDIR is shown in Figure 15-127 and described in Table 15-138.

Return to the Summary Table.

GPIO A Direction Register (GPIO0 to 31)
Controls direction of GPIO pins when the specified pin is configured in GPIO mode.
0: Configures pin as input.
1: Configures pin as output.
Reading the register returns the current value of the register setting.

Figure 15-127. GPBDIR Register

31	30	29	28	27	26	25	24
GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-138. GPBDIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO63	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
30	GPIO62	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
29	GPIO61	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
28	GPIO60	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
27	GPIO59	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
26	GPIO58	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
25	GPIO57	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
24	GPIO56	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
23	GPIO55	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
22	GPIO54	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
21	GPIO53	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn

Table 15-138. GPBDIR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO52	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
19	GPIO51	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
18	GPIO50	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
17	GPIO49	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
16	GPIO48	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
15	GPIO47	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
14	GPIO46	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
13	GPIO45	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
12	GPIO44	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
11	GPIO43	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
10	GPIO42	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
9	GPIO41	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
8	GPIO40	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
7	GPIO39	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
6	GPIO38	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
5	GPIO37	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
4	GPIO36	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
3	GPIO35	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
2	GPIO34	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
1	GPIO33	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
0	GPIO32	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn

15.11.3.11 GPCDAT Register (Offset = 28h) [Reset = 00000000h]

GPCDAT is shown in Figure 15-128 and described in Table 15-139.

Return to the Summary Table.

GPIO C Data Register (GPIO64 to 95)
Reading this register indicates the current status of the GPIO pin, irrespective of which mode the pin is in. Writing to this register will set the GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero.
DESIGNER NOTE:
[1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the

Figure 15-128. GPCDAT Register

31	30	29	28	27	26	25	24
GPIO95	GPIO94	GPIO93	GPIO92	GPIO91	GPIO90	GPIO89	GPIO88
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO87	GPIO86	GPIO85	GPIO84	GPIO83	GPIO82	GPIO81	GPIO80
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO79	GPIO78	GPIO77	GPIO76	GPIO75	GPIO74	GPIO73	GPIO72
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-139. GPCDAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO95	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
30	GPIO94	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
29	GPIO93	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
28	GPIO92	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
27	GPIO91	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
26	GPIO90	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
25	GPIO89	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
24	GPIO88	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
23	GPIO87	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
22	GPIO86	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn

Table 15-139. GPCDAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	GPIO85	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
20	GPIO84	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
19	GPIO83	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
18	GPIO82	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
17	GPIO81	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
16	GPIO80	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
15	GPIO79	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
14	GPIO78	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
13	GPIO77	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
12	GPIO76	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
11	GPIO75	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
10	GPIO74	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
9	GPIO73	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
8	GPIO72	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
7	GPIO71	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
6	GPIO70	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
5	GPIO69	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
4	GPIO68	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
3	GPIO67	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
2	GPIO66	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
1	GPIO65	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
0	GPIO64	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn

15.11.3.12 GPCSET Register (Offset = 2Ch) [Reset = 00000000h]

GPCSET is shown in Figure 15-129 and described in Table 15-140.

Return to the Summary Table.

GPIO C Data Set Register (GPIO64 to 95)
Writing a 1 will force GPIO output data latch to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-129. GPCSET Register

31	30	29	28	27	26	25	24
GPIO95	GPIO94	GPIO93	GPIO92	GPIO91	GPIO90	GPIO89	GPIO88
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
GPIO87	GPIO86	GPIO85	GPIO84	GPIO83	GPIO82	GPIO81	GPIO80
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
GPIO79	GPIO78	GPIO77	GPIO76	GPIO75	GPIO74	GPIO73	GPIO72
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-140. GPCSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO95	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
30	GPIO94	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
29	GPIO93	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
28	GPIO92	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
27	GPIO91	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
26	GPIO90	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
25	GPIO89	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
24	GPIO88	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
23	GPIO87	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
22	GPIO86	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
21	GPIO85	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
20	GPIO84	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn

Table 15-140. GPCSET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO83	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
18	GPIO82	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
17	GPIO81	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
16	GPIO80	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
15	GPIO79	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
14	GPIO78	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
13	GPIO77	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
12	GPIO76	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
11	GPIO75	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
10	GPIO74	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
9	GPIO73	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
8	GPIO72	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
7	GPIO71	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
6	GPIO70	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
5	GPIO69	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
4	GPIO68	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
3	GPIO67	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
2	GPIO66	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
1	GPIO65	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
0	GPIO64	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn

15.11.3.13 GPCCLEAR Register (Offset = 30h) [Reset = 00000000h]

GPCCLEAR is shown in Figure 15-130 and described in Table 15-141.

Return to the Summary Table.

GPIO C Data Clear Register (GPIO64 to 95)
Writing a 1 will force GPIO0 output data latch to 0.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-130. GPCCLEAR Register

31	30	29	28	27	26	25	24
GPIO95	GPIO94	GPIO93	GPIO92	GPIO91	GPIO90	GPIO89	GPIO88
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
GPIO87	GPIO86	GPIO85	GPIO84	GPIO83	GPIO82	GPIO81	GPIO80
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
GPIO79	GPIO78	GPIO77	GPIO76	GPIO75	GPIO74	GPIO73	GPIO72
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-141. GPCCLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO95	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
30	GPIO94	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
29	GPIO93	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
28	GPIO92	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
27	GPIO91	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
26	GPIO90	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
25	GPIO89	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
24	GPIO88	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
23	GPIO87	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
22	GPIO86	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
21	GPIO85	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
20	GPIO84	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn

Table 15-141. GPCCLEAR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO83	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
18	GPIO82	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
17	GPIO81	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
16	GPIO80	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
15	GPIO79	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
14	GPIO78	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
13	GPIO77	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
12	GPIO76	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
11	GPIO75	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
10	GPIO74	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
9	GPIO73	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
8	GPIO72	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
7	GPIO71	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
6	GPIO70	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
5	GPIO69	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
4	GPIO68	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
3	GPIO67	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
2	GPIO66	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
1	GPIO65	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
0	GPIO64	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn

15.11.3.14 GPCTOGGLE Register (Offset = 34h) [Reset = 00000000h]

GPCTOGGLE is shown in Figure 15-131 and described in Table 15-142.

Return to the Summary Table.

GPIO C Data Toggle Register (GPIO64 to 95)
Writing a 1 will toggle GPIO0 output data latch 1 to 0 or 0 to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-131. GPCTOGGLE Register

31	30	29	28	27	26	25	24
GPIO95	GPIO94	GPIO93	GPIO92	GPIO91	GPIO90	GPIO89	GPIO88
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
GPIO87	GPIO86	GPIO85	GPIO84	GPIO83	GPIO82	GPIO81	GPIO80
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
GPIO79	GPIO78	GPIO77	GPIO76	GPIO75	GPIO74	GPIO73	GPIO72
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-142. GPCTOGGLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO95	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
30	GPIO94	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
29	GPIO93	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
28	GPIO92	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
27	GPIO91	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
26	GPIO90	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
25	GPIO89	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
24	GPIO88	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
23	GPIO87	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
22	GPIO86	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
21	GPIO85	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
20	GPIO84	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn

Table 15-142. GPCTOGGLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO83	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
18	GPIO82	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
17	GPIO81	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
16	GPIO80	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
15	GPIO79	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
14	GPIO78	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
13	GPIO77	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
12	GPIO76	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
11	GPIO75	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
10	GPIO74	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
9	GPIO73	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
8	GPIO72	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
7	GPIO71	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
6	GPIO70	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
5	GPIO69	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
4	GPIO68	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
3	GPIO67	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
2	GPIO66	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
1	GPIO65	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
0	GPIO64	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn

15.11.3.15 GPCDIR Register (Offset = 38h) [Reset = 00000000h]

GPCDIR is shown in Figure 15-132 and described in Table 15-143.

Return to the Summary Table.

GPIO A Direction Register (GPIO0 to 31)
Controls direction of GPIO pins when the specified pin is configured in GPIO mode.
0: Configures pin as input.
1: Configures pin as output.
Reading the register returns the current value of the register setting.

Figure 15-132. GPCDIR Register

31	30	29	28	27	26	25	24
GPIO95	GPIO94	GPIO93	GPIO92	GPIO91	GPIO90	GPIO89	GPIO88
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO87	GPIO86	GPIO85	GPIO84	GPIO83	GPIO82	GPIO81	GPIO80
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO79	GPIO78	GPIO77	GPIO76	GPIO75	GPIO74	GPIO73	GPIO72
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-143. GPCDIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO95	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
30	GPIO94	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
29	GPIO93	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
28	GPIO92	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
27	GPIO91	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
26	GPIO90	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
25	GPIO89	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
24	GPIO88	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
23	GPIO87	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
22	GPIO86	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
21	GPIO85	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn

Table 15-143. GPCDIR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO84	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
19	GPIO83	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
18	GPIO82	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
17	GPIO81	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
16	GPIO80	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
15	GPIO79	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
14	GPIO78	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
13	GPIO77	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
12	GPIO76	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
11	GPIO75	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
10	GPIO74	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
9	GPIO73	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
8	GPIO72	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
7	GPIO71	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
6	GPIO70	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
5	GPIO69	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
4	GPIO68	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
3	GPIO67	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
2	GPIO66	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
1	GPIO65	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
0	GPIO64	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn

15.11.3.16 GPDDAT Register (Offset = 3Ch) [Reset = 00000000h]

GPDDAT is shown in Figure 15-133 and described in Table 15-144.

Return to the Summary Table.

GPIO D Data Register (GPIO96 to 127)
Reading this register indicates the current status of the GPIO pin, irrespective of which mode the pin is in. Writing to this register will set the GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero.
DESIGNER NOTE:
[1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register.

Figure 15-133. GPDDAT Register

31	30	29	28	27	26	25	24
GPIO127	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO105	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO103	RESERVED	GPIO101	GPIO100	GPIO99	GPIO98	GPIO97	GPIO96
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-144. GPDDAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO127	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved

Table 15-144. GPDDAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	GPIO105	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
8	RESERVED	R/W	0h	Reserved
7	GPIO103	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
6	RESERVED	R/W	0h	Reserved
5	GPIO101	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
4	GPIO100	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
3	GPIO99	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
2	GPIO98	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
1	GPIO97	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
0	GPIO96	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn

15.11.3.17 GPDSET Register (Offset = 40h) [Reset = 00000000h]

GPDSET is shown in Figure 15-134 and described in Table 15-145.

Return to the Summary Table.

GPIO D Data Set Register (GPIO96 to 127)
Writing a 1 will force GPIO output data latch to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-134. GPDSET Register

31	30	29	28	27	26	25	24
GPIO127	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO105	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO103	RESERVED	GPIO101	GPIO100	GPIO99	GPIO98	GPIO97	GPIO96
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-145. GPDSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO127	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
30	RESERVED	R-0/W	0h	Reserved
29	RESERVED	R-0/W	0h	Reserved
28	RESERVED	R-0/W	0h	Reserved
27	RESERVED	R-0/W	0h	Reserved
26	RESERVED	R-0/W	0h	Reserved
25	RESERVED	R-0/W	0h	Reserved
24	RESERVED	R-0/W	0h	Reserved
23	RESERVED	R-0/W	0h	Reserved
22	RESERVED	R-0/W	0h	Reserved
21	RESERVED	R-0/W	0h	Reserved
20	RESERVED	R-0/W	0h	Reserved
19	RESERVED	R-0/W	0h	Reserved
18	RESERVED	R-0/W	0h	Reserved
17	RESERVED	R-0/W	0h	Reserved
16	RESERVED	R-0/W	0h	Reserved
15	RESERVED	R-0/W	0h	Reserved
14	RESERVED	R-0/W	0h	Reserved
13	RESERVED	R-0/W	0h	Reserved
12	RESERVED	R-0/W	0h	Reserved
11	RESERVED	R-0/W	0h	Reserved
10	RESERVED	R-0/W	0h	Reserved

Table 15-145. GPDSET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	GPIO105	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
8	RESERVED	R-0/W	0h	Reserved
7	GPIO103	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
6	RESERVED	R-0/W	0h	Reserved
5	GPIO101	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
4	GPIO100	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
3	GPIO99	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
2	GPIO98	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
1	GPIO97	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn
0	GPIO96	R-0/W	0h	Output Set bit for this pin Reset type: CPUx.SYSRSn

15.11.3.18 GPDCLEAR Register (Offset = 44h) [Reset = 00000000h]

GPDCLEAR is shown in Figure 15-135 and described in Table 15-146.

Return to the Summary Table.

GPIO D Data Clear Register (GPIO96 to 127)
Writing a 1 will force GPIO0 output data latch to 0.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-135. GPDCLEAR Register

31	30	29	28	27	26	25	24
GPIO127	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO105	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO103	RESERVED	GPIO101	GPIO100	GPIO99	GPIO98	GPIO97	GPIO96
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-146. GPDCLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO127	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
30	RESERVED	R-0/W	0h	Reserved
29	RESERVED	R-0/W	0h	Reserved
28	RESERVED	R-0/W	0h	Reserved
27	RESERVED	R-0/W	0h	Reserved
26	RESERVED	R-0/W	0h	Reserved
25	RESERVED	R-0/W	0h	Reserved
24	RESERVED	R-0/W	0h	Reserved
23	RESERVED	R-0/W	0h	Reserved
22	RESERVED	R-0/W	0h	Reserved
21	RESERVED	R-0/W	0h	Reserved
20	RESERVED	R-0/W	0h	Reserved
19	RESERVED	R-0/W	0h	Reserved
18	RESERVED	R-0/W	0h	Reserved
17	RESERVED	R-0/W	0h	Reserved
16	RESERVED	R-0/W	0h	Reserved
15	RESERVED	R-0/W	0h	Reserved
14	RESERVED	R-0/W	0h	Reserved
13	RESERVED	R-0/W	0h	Reserved
12	RESERVED	R-0/W	0h	Reserved
11	RESERVED	R-0/W	0h	Reserved
10	RESERVED	R-0/W	0h	Reserved

Table 15-146. GPDCLEAR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	GPIO105	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
8	RESERVED	R-0/W	0h	Reserved
7	GPIO103	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
6	RESERVED	R-0/W	0h	Reserved
5	GPIO101	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
4	GPIO100	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
3	GPIO99	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
2	GPIO98	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
1	GPIO97	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn
0	GPIO96	R-0/W	0h	Output Clear bit for this pin Reset type: CPUx.SYSRSn

15.11.3.19 GPDTOGGLE Register (Offset = 48h) [Reset = 00000000h]

GPDTOGGLE is shown in Figure 15-136 and described in Table 15-147.

Return to the Summary Table.

GPIO D Data Toggle Register (GPIO96 to 127)
Writing a 1 will toggle GPIO0 output data latch 1 to 0 or 0 to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-136. GPDTOGGLE Register

31	30	29	28	27	26	25	24
GPIO127	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO105	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO103	RESERVED	GPIO101	GPIO100	GPIO99	GPIO98	GPIO97	GPIO96
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-147. GPDTOGGLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO127	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
30	RESERVED	R-0/W	0h	Reserved
29	RESERVED	R-0/W	0h	Reserved
28	RESERVED	R-0/W	0h	Reserved
27	RESERVED	R-0/W	0h	Reserved
26	RESERVED	R-0/W	0h	Reserved
25	RESERVED	R-0/W	0h	Reserved
24	RESERVED	R-0/W	0h	Reserved
23	RESERVED	R-0/W	0h	Reserved
22	RESERVED	R-0/W	0h	Reserved
21	RESERVED	R-0/W	0h	Reserved
20	RESERVED	R-0/W	0h	Reserved
19	RESERVED	R-0/W	0h	Reserved
18	RESERVED	R-0/W	0h	Reserved
17	RESERVED	R-0/W	0h	Reserved
16	RESERVED	R-0/W	0h	Reserved
15	RESERVED	R-0/W	0h	Reserved
14	RESERVED	R-0/W	0h	Reserved
13	RESERVED	R-0/W	0h	Reserved
12	RESERVED	R-0/W	0h	Reserved
11	RESERVED	R-0/W	0h	Reserved
10	RESERVED	R-0/W	0h	Reserved

Table 15-147. GPDTOGGLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	GPIO105	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
8	RESERVED	R-0/W	0h	Reserved
7	GPIO103	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
6	RESERVED	R-0/W	0h	Reserved
5	GPIO101	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
4	GPIO100	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
3	GPIO99	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
2	GPIO98	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
1	GPIO97	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn
0	GPIO96	R-0/W	0h	Output Toggle Register GPIO pin Reset type: CPUx.SYSRSn

15.11.3.20 GPDDIR Register (Offset = 4Ch) [Reset = 00000000h]

GPDDIR is shown in Figure 15-137 and described in Table 15-148.

Return to the Summary Table.

GPIO A Direction Register (GPIO0 to 31)
Controls direction of GPIO pins when the specified pin is configured in GPIO mode.
0: Configures pin as input.
1: Configures pin as output.
Reading the register returns the current value of the register setting.

Figure 15-137. GPDDIR Register

31	30	29	28	27	26	25	24
GPIO127	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO105	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO103	RESERVED	GPIO101	GPIO100	GPIO99	GPIO98	GPIO97	GPIO96
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-148. GPDDIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO127	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved

Table 15-148. GPDDIR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RESERVED	R/W	0h	Reserved
9	GPIO105	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
8	RESERVED	R/W	0h	Reserved
7	GPIO103	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
6	RESERVED	R/W	0h	Reserved
5	GPIO101	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
4	GPIO100	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
3	GPIO99	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
2	GPIO98	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
1	GPIO97	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
0	GPIO96	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn

15.11.3.21 GPFDAT Register (Offset = 64h) [Reset = 00000000h]

GPFDAT is shown in Figure 15-138 and described in Table 15-149.

Return to the Summary Table.

GPIO F Data Register (GPIO160 to 191)
Reading this register indicates the current status of the GPIO pin, irrespective of which mode the pin is in. Writing to this register will set the GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero.
DESIGNER NOTE:
[1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the

Figure 15-138. GPFDAT Register

31	30	29	28	27	26	25	24
GPIO191	GPIO190	GPIO189	GPIO188	GPIO187	GPIO186	GPIO185	GPIO184
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO183	GPIO182	GPIO181	GPIO180	GPIO179	GPIO178	GPIO177	GPIO176
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO175	GPIO174	GPIO173	GPIO172	GPIO171	GPIO170	GPIO169	GPIO168
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO167	GPIO166	GPIO165	GPIO164	GPIO163	GPIO162	GPIO161	GPIO160
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-149. GPFDAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO191	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
30	GPIO190	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
29	GPIO189	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
28	GPIO188	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
27	GPIO187	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
26	GPIO186	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
25	GPIO185	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
24	GPIO184	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
23	GPIO183	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
22	GPIO182	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn

Table 15-149. GPFDAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	GPIO181	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
20	GPIO180	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
19	GPIO179	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
18	GPIO178	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
17	GPIO177	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
16	GPIO176	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
15	GPIO175	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
14	GPIO174	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
13	GPIO173	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
12	GPIO172	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
11	GPIO171	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
10	GPIO170	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
9	GPIO169	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
8	GPIO168	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
7	GPIO167	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
6	GPIO166	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
5	GPIO165	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
4	GPIO164	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
3	GPIO163	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
2	GPIO162	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
1	GPIO161	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn
0	GPIO160	R/W	0h	Data Register for this pin Reset type: CPUx.SYSRSn

15.11.3.22 GPGDAT Register (Offset = 78h) [Reset = 00000000h]

GPGDAT is shown in Figure 15-139 and described in Table 15-150.

Return to the Summary Table.

GPIO G Data Register (GPIO192 to 223)
Reading this register indicates the current status of the GPIO pin, irrespective of which mode the pin is in. Writing to this register will set the GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero.
DESIGNER NOTE:
[1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the

Figure 15-139. GPGDAT Register

31	30	29	28	27	26	25	24
GPIO223	GPIO222	GPIO221	GPIO220	GPIO219	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	GPIO213	GPIO212	GPIO211	GPIO210	GPIO209	GPIO208
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO207	GPIO206	GPIO205	GPIO204	GPIO203	GPIO202	GPIO201	GPIO200
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO199	GPIO198	GPIO197	GPIO196	GPIO195	GPIO194	GPIO193	GPIO192
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-150. GPGDAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO223	R/W	0h	Data Register for this pin Reset type: SYSRSn
30	GPIO222	R/W	0h	Data Register for this pin Reset type: SYSRSn
29	GPIO221	R/W	0h	Data Register for this pin Reset type: SYSRSn
28	GPIO220	R/W	0h	Data Register for this pin Reset type: SYSRSn
27	GPIO219	R/W	0h	Data Register for this pin Reset type: SYSRSn
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	GPIO213	R/W	0h	Data Register for this pin Reset type: SYSRSn
20	GPIO212	R/W	0h	Data Register for this pin Reset type: SYSRSn

Table 15-150. GPGDAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO211	R/W	0h	Data Register for this pin Reset type: SYSRSn
18	GPIO210	R/W	0h	Data Register for this pin Reset type: SYSRSn
17	GPIO209	R/W	0h	Data Register for this pin Reset type: SYSRSn
16	GPIO208	R/W	0h	Data Register for this pin Reset type: SYSRSn
15	GPIO207	R/W	0h	Data Register for this pin Reset type: SYSRSn
14	GPIO206	R/W	0h	Data Register for this pin Reset type: SYSRSn
13	GPIO205	R/W	0h	Data Register for this pin Reset type: SYSRSn
12	GPIO204	R/W	0h	Data Register for this pin Reset type: SYSRSn
11	GPIO203	R/W	0h	Data Register for this pin Reset type: SYSRSn
10	GPIO202	R/W	0h	Data Register for this pin Reset type: SYSRSn
9	GPIO201	R/W	0h	Data Register for this pin Reset type: SYSRSn
8	GPIO200	R/W	0h	Data Register for this pin Reset type: SYSRSn
7	GPIO199	R/W	0h	Data Register for this pin Reset type: SYSRSn
6	GPIO198	R/W	0h	Data Register for this pin Reset type: SYSRSn
5	GPIO197	R/W	0h	Data Register for this pin Reset type: SYSRSn
4	GPIO196	R/W	0h	Data Register for this pin Reset type: SYSRSn
3	GPIO195	R/W	0h	Data Register for this pin Reset type: SYSRSn
2	GPIO194	R/W	0h	Data Register for this pin Reset type: SYSRSn
1	GPIO193	R/W	0h	Data Register for this pin Reset type: SYSRSn
0	GPIO192	R/W	0h	Data Register for this pin Reset type: SYSRSn

15.11.3.23 GPGSET Register (Offset = 7Ch) [Reset = 00000000h]

GPGSET is shown in Figure 15-140 and described in Table 15-151.

Return to the Summary Table.

GPIO G Data Set Register (GPIO192 to 223)
Writing a 1 will force GPIO output data latch to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-140. GPGSET Register

31	30	29	28	27	26	25	24
GPIO223	GPIO222	GPIO221	GPIO220	GPIO219	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-151. GPGSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO223	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
30	GPIO222	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
29	GPIO221	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
28	GPIO220	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
27	GPIO219	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
26	RESERVED	R-0/W	0h	Reserved
25	RESERVED	R-0/W	0h	Reserved
24	RESERVED	R-0/W	0h	Reserved
23	RESERVED	R-0/W	0h	Reserved
22	RESERVED	R-0/W	0h	Reserved
21	RESERVED	R-0/W	0h	Reserved
20	RESERVED	R-0/W	0h	Reserved
19	RESERVED	R-0/W	0h	Reserved
18	RESERVED	R-0/W	0h	Reserved
17	RESERVED	R-0/W	0h	Reserved
16	RESERVED	R-0/W	0h	Reserved
15	RESERVED	R-0/W	0h	Reserved
14	RESERVED	R-0/W	0h	Reserved

Table 15-151. GPGSET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	RESERVED	R-0/W	0h	Reserved
12	RESERVED	R-0/W	0h	Reserved
11	RESERVED	R-0/W	0h	Reserved
10	RESERVED	R-0/W	0h	Reserved
9	RESERVED	R-0/W	0h	Reserved
8	RESERVED	R-0/W	0h	Reserved
7	RESERVED	R-0/W	0h	Reserved
6	RESERVED	R-0/W	0h	Reserved
5	RESERVED	R-0/W	0h	Reserved
4	RESERVED	R-0/W	0h	Reserved
3	RESERVED	R-0/W	0h	Reserved
2	RESERVED	R-0/W	0h	Reserved
1	RESERVED	R-0/W	0h	Reserved
0	RESERVED	R-0/W	0h	Reserved

15.11.3.24 GPGCLEAR Register (Offset = 80h) [Reset = 00000000h]

GPGCLEAR is shown in Figure 15-141 and described in Table 15-152.

Return to the Summary Table.

GPIO G Data Clear Register (GPIO192 to 223)
Writing a 1 will force GPIO0 output data latch to 0.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-141. GPGCLEAR Register

31	30	29	28	27	26	25	24
GPIO223	GPIO222	GPIO221	GPIO220	GPIO219	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-152. GPGCLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO223	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
30	GPIO222	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
29	GPIO221	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
28	GPIO220	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
27	GPIO219	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
26	RESERVED	R-0/W	0h	Reserved
25	RESERVED	R-0/W	0h	Reserved
24	RESERVED	R-0/W	0h	Reserved
23	RESERVED	R-0/W	0h	Reserved
22	RESERVED	R-0/W	0h	Reserved
21	RESERVED	R-0/W	0h	Reserved
20	RESERVED	R-0/W	0h	Reserved
19	RESERVED	R-0/W	0h	Reserved
18	RESERVED	R-0/W	0h	Reserved
17	RESERVED	R-0/W	0h	Reserved
16	RESERVED	R-0/W	0h	Reserved
15	RESERVED	R-0/W	0h	Reserved
14	RESERVED	R-0/W	0h	Reserved

Table 15-152. GPGCLEAR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	RESERVED	R-0/W	0h	Reserved
12	RESERVED	R-0/W	0h	Reserved
11	RESERVED	R-0/W	0h	Reserved
10	RESERVED	R-0/W	0h	Reserved
9	RESERVED	R-0/W	0h	Reserved
8	RESERVED	R-0/W	0h	Reserved
7	RESERVED	R-0/W	0h	Reserved
6	RESERVED	R-0/W	0h	Reserved
5	RESERVED	R-0/W	0h	Reserved
4	RESERVED	R-0/W	0h	Reserved
3	RESERVED	R-0/W	0h	Reserved
2	RESERVED	R-0/W	0h	Reserved
1	RESERVED	R-0/W	0h	Reserved
0	RESERVED	R-0/W	0h	Reserved

15.11.3.25 GPGTOGGLE Register (Offset = 84h) [Reset = 00000000h]

GPGTOGGLE is shown in Figure 15-142 and described in Table 15-153.

Return to the Summary Table.

GPIO G Data Toggle Register (GPIO192 to 223)
Writing a 1 will toggle GPIO0 output data latch 1 to 0 or 0 to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-142. GPGTOGGLE Register

31	30	29	28	27	26	25	24
GPIO223	GPIO222	GPIO221	GPIO220	GPIO219	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-153. GPGTOGGLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO223	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
30	GPIO222	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
29	GPIO221	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
28	GPIO220	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
27	GPIO219	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
26	RESERVED	R-0/W	0h	Reserved
25	RESERVED	R-0/W	0h	Reserved
24	RESERVED	R-0/W	0h	Reserved
23	RESERVED	R-0/W	0h	Reserved
22	RESERVED	R-0/W	0h	Reserved
21	RESERVED	R-0/W	0h	Reserved
20	RESERVED	R-0/W	0h	Reserved
19	RESERVED	R-0/W	0h	Reserved
18	RESERVED	R-0/W	0h	Reserved
17	RESERVED	R-0/W	0h	Reserved
16	RESERVED	R-0/W	0h	Reserved
15	RESERVED	R-0/W	0h	Reserved
14	RESERVED	R-0/W	0h	Reserved

Table 15-153. GPGTOGGLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	RESERVED	R-0/W	0h	Reserved
12	RESERVED	R-0/W	0h	Reserved
11	RESERVED	R-0/W	0h	Reserved
10	RESERVED	R-0/W	0h	Reserved
9	RESERVED	R-0/W	0h	Reserved
8	RESERVED	R-0/W	0h	Reserved
7	RESERVED	R-0/W	0h	Reserved
6	RESERVED	R-0/W	0h	Reserved
5	RESERVED	R-0/W	0h	Reserved
4	RESERVED	R-0/W	0h	Reserved
3	RESERVED	R-0/W	0h	Reserved
2	RESERVED	R-0/W	0h	Reserved
1	RESERVED	R-0/W	0h	Reserved
0	RESERVED	R-0/W	0h	Reserved

15.11.3.26 GPGDIR Register (Offset = 88h) [Reset = 00000000h]

GPGDIR is shown in Figure 15-143 and described in Table 15-154.

Return to the Summary Table.

GPIO A Direction Register (GPIO0 to 31)
Controls direction of GPIO pins when the specified pin is configured in GPIO mode.
0: Configures pin as input.
1: Configures pin as output.
Reading the register returns the current value of the register setting.

Figure 15-143. GPGDIR Register

31	30	29	28	27	26	25	24
GPIO223	GPIO222	GPIO221	GPIO220	GPIO219	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-154. GPGDIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO223	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
30	GPIO222	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
29	GPIO221	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
28	GPIO220	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
27	GPIO219	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved

Table 15-154. GPGDIR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	RESERVED	R/W	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

15.11.3.27 GPHDAT Register (Offset = 8Ch) [Reset = 00000000h]

GPHDAT is shown in Figure 15-144 and described in Table 15-155.

Return to the Summary Table.

GPIO H Data Register (GPIO224 to 255)
Reading this register indicates the current status of the GPIO pin, irrespective of which mode the pin is in. Writing to this register will set the GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero.
DESIGNER NOTE:
[1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the

Figure 15-144. GPHDAT Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO249	GPIO248
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO247	GPIO246	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	GPIO240
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	GPIO236	GPIO235	GPIO234	GPIO233	GPIO232
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	GPIO229	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-155. GPHDAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	GPIO249	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn

Table 15-155. GPHDAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	GPIO248	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
23	GPIO247	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
22	GPIO246	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
21	GPIO245	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
20	GPIO244	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn

Table 15-155. GPHDAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO243	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
18	GPIO242	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
17	GPIO241	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
16	GPIO240	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
15	GPIO239	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn

Table 15-155. GPHDAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	GPIO238	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
13	GPIO237	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
12	GPIO236	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
11	GPIO235	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
10	GPIO234	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn

Table 15-155. GPHDAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	GPIO233	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
8	GPIO232	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
7	GPIO231	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
6	GPIO230	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
5	GPIO229	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn

Table 15-155. GPHDAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	GPIO228	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
3	GPIO227	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
2	GPIO226	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
1	GPIO225	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
0	GPIO224	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn

15.11.3.28 GPHSET Register (Offset = 90h) [Reset = 00000000h]

GPHSET is shown in Figure 15-145 and described in Table 15-156.

Return to the Summary Table.

GPIO H Data Set Register (GPIO224 to 255)
Writing a 1 will force GPIO output data latch to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-145. GPHSET Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO249	GPIO248
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
GPIO247	GPIO246	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	GPIO240
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	GPIO236	GPIO235	GPIO234	GPIO233	GPIO232
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	GPIO229	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-156. GPHSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W	0h	Reserved
30	RESERVED	R-0/W	0h	Reserved
29	RESERVED	R-0/W	0h	Reserved
28	RESERVED	R-0/W	0h	Reserved
27	RESERVED	R-0/W	0h	Reserved
26	RESERVED	R-0/W	0h	Reserved
25	GPIO249	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
24	GPIO248	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
23	GPIO247	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
22	GPIO246	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
21	GPIO245	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
20	GPIO244	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
19	GPIO243	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
18	GPIO242	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
17	GPIO241	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn

Table 15-156. GPHSET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	GPIO240	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
15	GPIO239	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
14	GPIO238	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
13	GPIO237	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
12	GPIO236	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
11	GPIO235	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
10	GPIO234	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
9	GPIO233	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
8	GPIO232	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
7	GPIO231	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
6	GPIO230	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
5	GPIO229	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
4	GPIO228	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
3	GPIO227	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
2	GPIO226	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
1	GPIO225	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
0	GPIO224	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn

15.11.3.29 GPHCLEAR Register (Offset = 94h) [Reset = 00000000h]

GPHCLEAR is shown in Figure 15-146 and described in Table 15-157.

Return to the Summary Table.

GPIO H Data Clear Register (GPIO224 to 255)
Writing a 1 will force GPIO0 output data latch to 0.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-146. GPHCLEAR Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO249	GPIO248
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
GPIO247	GPIO246	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	GPIO240
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	GPIO236	GPIO235	GPIO234	GPIO233	GPIO232
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	GPIO229	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-157. GPHCLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W	0h	Reserved
30	RESERVED	R-0/W	0h	Reserved
29	RESERVED	R-0/W	0h	Reserved
28	RESERVED	R-0/W	0h	Reserved
27	RESERVED	R-0/W	0h	Reserved
26	RESERVED	R-0/W	0h	Reserved
25	GPIO249	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
24	GPIO248	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
23	GPIO247	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
22	GPIO246	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
21	GPIO245	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
20	GPIO244	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
19	GPIO243	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
18	GPIO242	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
17	GPIO241	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn

Table 15-157. GPHCLEAR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	GPIO240	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
15	GPIO239	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
14	GPIO238	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
13	GPIO237	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
12	GPIO236	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
11	GPIO235	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
10	GPIO234	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
9	GPIO233	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
8	GPIO232	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
7	GPIO231	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
6	GPIO230	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
5	GPIO229	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
4	GPIO228	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
3	GPIO227	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
2	GPIO226	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
1	GPIO225	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
0	GPIO224	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn

15.11.3.30 GPHTOGGLE Register (Offset = 98h) [Reset = 00000000h]

GPHTOGGLE is shown in Figure 15-147 and described in Table 15-158.

Return to the Summary Table.

GPIO H Data Toggle Register (GPIO224 to 255)
Writing a 1 will toggle GPIO0 output data latch 1 to 0 or 0 to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 15-147. GPHTOGGLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO249	GPIO248
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
GPIO247	GPIO246	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	GPIO240
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	GPIO236	GPIO235	GPIO234	GPIO233	GPIO232
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	GPIO229	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 15-158. GPHTOGGLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W	0h	Reserved
30	RESERVED	R-0/W	0h	Reserved
29	RESERVED	R-0/W	0h	Reserved
28	RESERVED	R-0/W	0h	Reserved
27	RESERVED	R-0/W	0h	Reserved
26	RESERVED	R-0/W	0h	Reserved
25	GPIO249	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
24	GPIO248	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
23	GPIO247	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
22	GPIO246	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
21	GPIO245	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
20	GPIO244	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
19	GPIO243	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
18	GPIO242	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
17	GPIO241	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn

Table 15-158. GPHTOGGLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	GPIO240	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
15	GPIO239	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
14	GPIO238	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
13	GPIO237	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
12	GPIO236	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
11	GPIO235	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
10	GPIO234	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
9	GPIO233	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
8	GPIO232	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
7	GPIO231	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
6	GPIO230	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
5	GPIO229	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
4	GPIO228	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
3	GPIO227	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
2	GPIO226	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
1	GPIO225	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
0	GPIO224	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn

15.11.3.31 GPHDIR Register (Offset = 9Ch) [Reset = 00000000h]

GPHDIR is shown in Figure 15-148 and described in Table 15-159.

Return to the Summary Table.

GPIO A Direction Register (GPIO0 to 31)
Controls direction of GPIO pins when the specified pin is configured in GPIO mode.
0: Configures pin as input.
1: Configures pin as output.
Reading the register returns the current value of the register setting.

Figure 15-148. GPHDIR Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO249	GPIO248
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO247	GPIO246	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	GPIO240
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	GPIO236	GPIO235	GPIO234	GPIO233	GPIO232
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	GPIO229	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-159. GPHDIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	GPIO249	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
24	GPIO248	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
23	GPIO247	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
22	GPIO246	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
21	GPIO245	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
20	GPIO244	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
19	GPIO243	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
18	GPIO242	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn

Table 15-159. GPHDIR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	GPIO241	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
16	GPIO240	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
15	GPIO239	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
14	GPIO238	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
13	GPIO237	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
12	GPIO236	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
11	GPIO235	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
10	GPIO234	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
9	GPIO233	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
8	GPIO232	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
7	GPIO231	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
6	GPIO230	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
5	GPIO229	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
4	GPIO228	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
3	GPIO227	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
2	GPIO226	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
1	GPIO225	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn
0	GPIO224	R/W	0h	Defines direction for this pin in GPIO mode Reset type: CPUx.SYSRSn

15.11.4 GPIO_DATA_READ_REGS Registers

Table 15-160 lists the memory-mapped registers for the GPIO_DATA_READ_REGS registers. All register offset addresses not listed in Table 15-160 should be considered as reserved locations and the register contents should not be modified.

Table 15-160. GPIO_DATA_READ_REGS Registers

Offset	Acronym	Register Name	Protection
0h	GPADAT_R	GPIO A Data Read Register	
4h	GPBDAT_R	GPIO B Data Read Register	
8h	GPCDAT_R	GPIO C Data Read Register	
Ch	GPDDAT_R	GPIO D Data Read Register	
14h	GPFDAT_R	GPIO F Data Read Register	
18h	GPGDAT_R	GPIO G Data Read Register	
1Ch	GPHDAT_R	GPIO H Data Read Register	

Complex bit access types are encoded to fit into small table cells. Table 15-161 shows the codes that are used for access types in this section.

Table 15-161. GPIO_DATA_READ_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

15.11.4.1 GPADAT_R Register (Offset = 0h) [Reset = 00000000h]

GPADAT_R is shown in Figure 15-149 and described in Table 15-162.

Return to the Summary Table.

GPIO A Data Read Register.
Returns the contents of GPADAT register on a read, write to this register has no effect

Figure 15-149. GPADAT_R Register

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

Table 15-162. GPADAT_R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	A read from this register returns the contents of GPADAT register, writes have no impact Reset type: CPUx.SYSRSn

15.11.4.2 GPBDAT_R Register (Offset = 4h) [Reset = 00000000h]

GPBDAT_R is shown in Figure 15-150 and described in Table 15-163.

Return to the Summary Table.

GPIO B Data Read Register.
Returns the contents of GPBDAT register on a read, write to this register has no effect

Figure 15-150. GPBDAT_R Register

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

Table 15-163. GPBDAT_R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	A read from this register returns the contents of GPBDAT register, writes have no impact Reset type: CPUx.SYSRSn

15.11.4.3 GPCDAT_R Register (Offset = 8h) [Reset = 00000000h]

GPCDAT_R is shown in Figure 15-151 and described in Table 15-164.

Return to the Summary Table.

GPIO C Data Read Register.
Returns the contents of GPCDAT register on a read, write to this register has no effect

Figure 15-151. GPCDAT_R Register

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

Table 15-164. GPCDAT_R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	A read from this register returns the contents of GPCDAT register, writes have no impact Reset type: CPUx.SYSRSn

15.11.4.4 GPDDAT_R Register (Offset = Ch) [Reset = 00000000h]

GPDDAT_R is shown in Figure 15-152 and described in Table 15-165.

Return to the Summary Table.

GPIO D Data Read Register.
Returns the contents of GPDDAT register on a read, write to this register has no effect

Figure 15-152. GPDDAT_R Register

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

Table 15-165. GPDDAT_R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	A read from this register returns the contents of GPDDAT register, writes have no impact Reset type: CPUx.SYSRSn

15.11.4.5 GPFDAT_R Register (Offset = 14h) [Reset = 00000000h]

GPFDAT_R is shown in Figure 15-153 and described in Table 15-166.

Return to the Summary Table.

GPIO F Data Read Register.
Returns the contents of GPFDAT register on a read, write to this register has no effect

Figure 15-153. GPFDAT_R Register

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

Table 15-166. GPFDAT_R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	A read from this register returns the contents of GPFDAT register, writes have no impact Reset type: CPUx.SYSRSn

15.11.4.6 GPGDAT_R Register (Offset = 18h) [Reset = 00000000h]

GPGDAT_R is shown in Figure 15-154 and described in Table 15-167.

Return to the Summary Table.

GPIO G Data Read Register.
Returns the contents of GPGDAT register on a read, write to this register has no effect

Figure 15-154. GPGDAT_R Register

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

Table 15-167. GPGDAT_R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	A read from this register returns the contents of GPGDAT register, writes have no impact Reset type: CPUx.SYSRSn

15.11.4.7 GPHDAT_R Register (Offset = 1Ch) [Reset = 00000000h]

GPHDAT_R is shown in Figure 15-155 and described in Table 15-168.

Return to the Summary Table.

GPIO H Data Read Register.
Returns the contents of GPHDAT register on a read, write to this register has no effect

Figure 15-155. GPHDAT_R Register

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

Table 15-168. GPHDAT_R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	A read from this register returns the contents of GPHDAT register, writes have no impact Reset type: CPUx.SYSRSn

Chapter 16

Interprocessor Communication (IPC)

The Interprocessor Communications (IPC) module allows communication between the two CPU subsystems.

16.1 Introduction

This section details the IPC features that each CPU can use to request and share information. The IPC features are:

- IPC flags and interrupts
- IPC command registers
- Free-running counter

All IPC features are independent of each other, and most do not require any specific data format.

IPC Module Architecture shows the design structure of the IPC module. The functionality is the same between any two CPUs.

There is no message RAM for devices with C29x processors, since it is possible to designate any memory as readable or writable by the various CPUs.

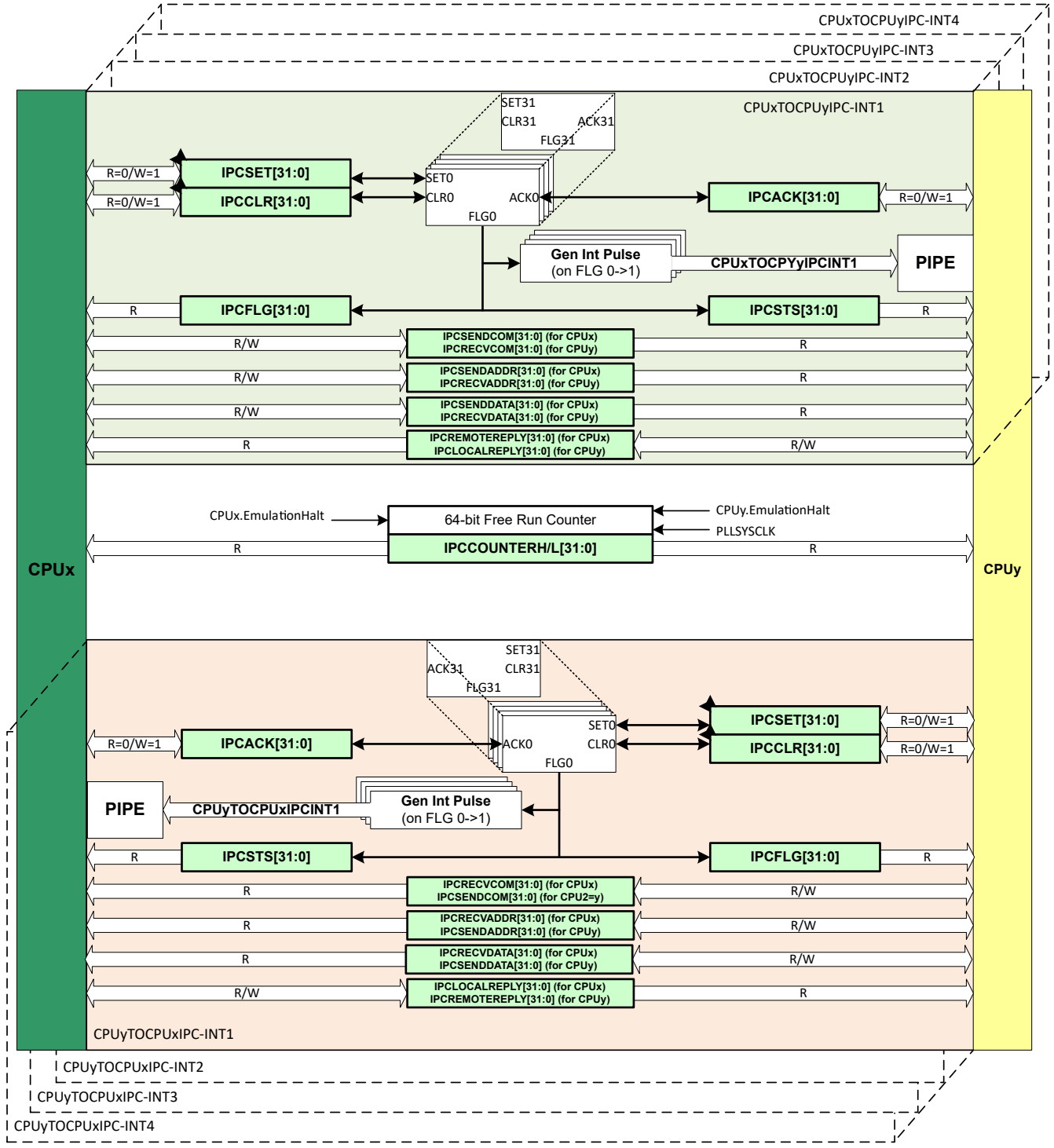


Figure 16-1. IPC Module Architecture

16.2 IPC Flags and Interrupts

There are 32 IPC event signals in each direction between the CPU pairs. These signals can be used for flag-based event polling. Between any two C29x cores, each of the four blocks of IPC registers has one IPC interrupt to go in each direction. This IPC interrupt can only be triggered by IPC flag 0.

16.3 IPC Command Registers

The IPC command registers provide a simple and flexible way for the CPUs to exchange more complex messages. Each CPU has eight dedicated registers; four for sending messages and four for receiving messages. The register names were chosen to support a simple command/response protocol, but can be used for any purpose. Only the read/write permissions are determined by hardware; the data format is entirely software-defined.

For sending messages, each CPU has three writable registers and one read-only register. Those same registers are accessible on the remote CPU as three read-only registers and one writable register. Table 16-1 shows the command registers.

Table 16-1. IPC Command Registers

Local Register Name	Local CPU	Remote CPU	Remote Register Name
IPCSENDCOM	R/W	R	IPCRECVCOM
IPCSENDADDR	R/W	R	IPCRECVADDR
IPCSENDDATA	R/W	R	IPCRECVDATA
IPCREMOTEREPLY	R	R/W	IPCLOCALREPLY

16.4 Free-Running Counter

A 64-bit free-running counter is present in the device and can be used to timestamp IPC events between processors. The counter is clocked by PLLSYSCLK and reset by XRSn. The counter is implemented as two 32-bit registers, IPCCOUNTERH and IPCCOUNTERL. When IPCCOUNTERL is read, the value of IPCCOUNTERH is saved. A subsequent read to IPCCOUNTERH returns this saved value. Therefore, the user must always read IPCCOUNTERL first then read IPCCOUNTERH next. This design prevents race conditions due to IPCCOUNTERL overflowing between reads of the two registers.

The free-running counter stops only when emulation is suspended (when debugger hits a breakpoint) on all CPUs. If any core is executing, the counter runs.

16.5 IPC Communication Protocol

This section describes the hardware support options for IPC communication between the two CPUs. These options can be used independently or in combination. All flag definitions and data formats are entirely user-defined.

- The flag system supports event-based communication via interrupts and register polling.
  - CPUx can raise an IPC event by writing to any of the 32 bits of the IPCSET register. This sets the corresponding bits in the CPUx IPCFLG register and CPUy IPCSTS register.
  - CPUy can signal the response to the event by setting the appropriate bit in the IPCACK register. This clears the corresponding bits in the CPUx IPCFLG register and the CPUy IPCSTS register.
  - If CPUx needs to cancel an event, CPUx can set the appropriate bit in the IPCCLR register. This has the same effect as CPUy writing to IPCACK.
  - Flag 0 (set using IPCSET[0]) fires an interrupt to the remote CPU. The remote CPU must configure the PIPE module properly to receive an IPC interrupt. Flags 1–31 (set using IPCSET[31:1]) do not produce interrupts. Multiple flags can be set, acknowledged, and cleared simultaneously.
- The command registers support sending several distinct pieces of information and are named COM, ADDR, DATA, and REPLY for convenience only and can hold whatever data the application needs.
  - CPUx can write data to the IPCSENDCOM, IPCSENDADDR, and IPCSENDDATA registers. CPUy receives these in the IPCRECVCOM, IPCRECVADDR, and IPCRECVDATA registers.
  - CPUy can respond by writing to the IPCLOCALREPLY register. CPUx receives this data in the IPCREMOTEREPLY register.

16.6 Software

16.6.1 IPC Registers to Driverlib Functions

Table 16-2. IPC Registers to Driverlib Functions

File	Driverlib Function
COUNTERL	
ipc.h	IPC_getCounter
COUNTERH	
ipc.h	IPC_getCounter
CPU1TOCPU2INTIPCSET(i)	
-	
CPU1TOCPU2INTIPCCLR(i)	
-	
CPU1TOCPU2INTIPCFLG(i)	
-	
CPU1TOCPU2INTIPCSENDCOM(i)	
-	
CPU1TOCPU2INTIPCSENDADDR(i)	
-	
CPU1TOCPU2INTIPCSENDDATA(i)	
-	
CPU2TOCPU1INTREMOTEREPLY(i)	
-	
CPU1TOCPU3INTIPCSET(i)	
-	
CPU1TOCPU3INTIPCCLR(i)	
-	
CPU1TOCPU3INTIPCFLG(i)	
-	
CPU1TOCPU3INTIPCSENDCOM(i)	
-	
CPU1TOCPU3INTIPCSENDADDR(i)	
-	
CPU1TOCPU3INTIPCSENDDATA(i)	
-	
CPU3TOCPU1INTREMOTEREPLY(i)	
-	
CPU1TOHSMINTIPCSET(i)	
-	
CPU1TOHSMINTIPCCLR(i)	
-	
CPU1TOHSMINTIPCFLG(i)	
-	
CPU2TOCPU1INTIPCSTS(i)	
-	
CPU1TOCPU2INTIPCACK(i)	
-	

Table 16-2. IPC Registers to Driverlib Functions (continued)

File	Driverlib Function
CPU2TOCPU1INTIPCRECVCOM(i)	
-	
CPU2TOCPU1INTIPCRECVADDR(i)	
-	
CPU2TOCPU1INTIPCRECVDATA(i)	
-	
CPU1TOCPU2INTLOCALREPLY(i)	
-	
CPU3TOCPU1INTIPCSTS(i)	
-	
CPU1TOCPU3INTIPCACK(i)	
-	
CPU3TOCPU1INTIPCRECVCOM(i)	
-	
CPU3TOCPU1INTIPCRECVADDR(i)	
-	
CPU3TOCPU1INTIPCRECVDATA(i)	
-	
CPU1TOCPU3INTLOCALREPLY(i)	
-	
CPU2TOCPU1INTIPCSET(i)	
-	
CPU2TOCPU1INTIPCCLR(i)	
-	
CPU2TOCPU1INTIPCFLG(i)	
-	
CPU2TOCPU1INTIPCSENDCOM(i)	
-	
CPU2TOCPU1INTIPCSENDADDR(i)	
-	
CPU2TOCPU1INTIPCSENDDATA(i)	
-	
CPU1TOCPU2INTREMOTEREPLY(i)	
-	
CPU2TOCPU3INTIPCSET(i)	
-	
CPU2TOCPU3INTIPCCLR(i)	
-	
CPU2TOCPU3INTIPCFLG(i)	
-	
CPU2TOCPU3INTIPCSENDCOM(i)	
-	
CPU2TOCPU3INTIPCSENDADDR(i)	
-	
CPU2TOCPU3INTIPCSENDDATA(i)	

Table 16-2. IPC Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
CPU3TOCPU2INTREMOTEREPLY(i)	
-	
CPU2TOHSMINTIPCSET(i)	
-	
CPU2TOHSMINTIPCCLR(i)	
-	
CPU2TOHSMINTIPCFLG(i)	
-	
CPU1TOCPU2INTIPCSTS(i)	
-	
CPU2TOCPU1INTIPCACK(i)	
-	
CPU1TOCPU2INTIPCRECVCOM(i)	
-	
CPU1TOCPU2INTIPCRECVADDR(i)	
-	
CPU1TOCPU2INTIPCRECVDATA(i)	
-	
CPU2TOCPU1INTLOCALREPLY(i)	
-	
CPU3TOCPU2INTIPCSTS(i)	
-	
CPU2TOCPU3INTIPCACK(i)	
-	
CPU3TOCPU2INTIPCRECVCOM(i)	
-	
CPU3TOCPU2INTIPCRECVADDR(i)	
-	
CPU3TOCPU2INTIPCRECVDATA(i)	
-	
CPU2TOCPU3INTLOCALREPLY(i)	
-	
CPU3TOCPU1INTIPCSET(i)	
-	
CPU3TOCPU1INTIPCCLR(i)	
-	
CPU3TOCPU1INTIPCFLG(i)	
-	
CPU3TOCPU1INTIPCSENDCOM(i)	
-	
CPU3TOCPU1INTIPCSENDADDR(i)	
-	
CPU3TOCPU1INTIPCSENDDATA(i)	
-	

Table 16-2. IPC Registers to Driverlib Functions (continued)

File	Driverlib Function
CPU1TOCPU3INTREMOTEREPLY(i)	
-	
CPU3TOCPU2INTIPCSET(i)	
-	
CPU3TOCPU2INTIPCCLR(i)	
-	
CPU3TOCPU2INTIPCFLG(i)	
-	
CPU3TOCPU2INTIPCSENDCOM(i)	
-	
CPU3TOCPU2INTIPCSENDADDR(i)	
-	
CPU3TOCPU2INTIPCSENDDATA(i)	
-	
CPU2TOCPU3INTREMOTEREPLY(i)	
-	
CPU3TOHSMINTIPCSET(i)	
-	
CPU3TOHSMINTIPCCLR(i)	
-	
CPU3TOHSMINTIPCFLG(i)	
-	
CPU1TOCPU3INTIPCSTS(i)	
-	
CPU3TOCPU1INTIPCACK(i)	
-	
CPU1TOCPU3INTIPCRECVCOM(i)	
-	
CPU1TOCPU3INTIPCRECVADDR(i)	
-	
CPU1TOCPU3INTIPCRECVDATA(i)	
-	
CPU3TOCPU1INTLOCALREPLY(i)	
-	
CPU2TOCPU3INTIPCSTS(i)	
-	
CPU3TOCPU2INTIPCACK(i)	
-	
CPU2TOCPU3INTIPCRECVCOM(i)	
-	
CPU2TOCPU3INTIPCRECVADDR(i)	
-	
CPU2TOCPU3INTIPCRECVDATA(i)	
-	
CPU3TOCPU2INTLOCALREPLY(i)	

Table 16-2. IPC Registers to Driverlib Functions (continued)

File	Driverlib Function
-	

16.6.2 IPC Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/ipc

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

16.6.2.1 IPC basic message passing example with interrupt - MULTI_CORE

FILE: ipc_ex1_basic_cpu1_cpu2_multi_c29x1.c

This example demonstrates how to configure IPC and pass information from C29x1 to C29x2 core without message queues.

When using CCS for debugging this Multi-core example, after launching the debug session,
- Connect to CPU1 and load only the c29x1.out.
- After the program is loaded, run CPU1.
- Once c29x1 disables lock-step and releases CPU2 out of reset, the program stops.
- Connect to CPU2 target now. c29x2.out would have started execution as soon as it is released from reset.
- In case of RAM configuration, restart the CPU2 target and load the symbols.

In FLASH configuration, the CPU2 application code is loaded to CPU1 FLASH. At runtime, CPU1 copies it to RAM for CPU2 to execute. *External Connections*
- None.

Watch Variables
- pass

16.6.2.2 IPC basic message passing example with interrupt - MULTI_CORE

FILE: ipc_ex1_basic_cpu1_cpu2_multi_c29x2.c

This example demonstrates how to configure IPC and pass information from C29x1 to C29x2 core without message queues.

When using CCS for debugging this Multi-core example, after launching the debug session,
- Connect to CPU1 and load the c29x2.out first. (In case of FLASH configuration, this might not halt at main, ignore this)
- Load c29x1.out next.
- After the program is loaded, run CPU1.
- Once c29x1 disables lock-step and releases CPU2 out of reset, the program stops.
- Connect to CPU2 target now. c29x2.out would have started execution as soon as it is released from reset.
- In case of RAM configuration, restart the CPU2 target and load the symbols.
- In case of FLASH, do a Power-on-Reset, connect to CPU1/ CPU2 and load the respective symbols.

In FLASH configuration, the CPU2 application code is loaded to CPU1 FLASH. At runtime, CPU1 copies it to RAM for CPU2 to execute. *External Connections*
- None.

Watch Variables
- None.

16.6.2.3 IPC basic message passing example with interrupt - MULTI_CORE

FILE: ipc_ex2_basic_cpu1_cpu3_multi_c29x1.c

This example demonstrates how to configure IPC and pass information from C29x1 to C29x3 core without message queues.

When using CCS for debugging this Multi-core example, after launching the debug session,
- Connect to CPU1 and load only the c29x1.out.
- After the program is loaded, run CPU1.
- Once c29x1 configures and releases CPU3 out of reset, the program stops.
- Connect to CPU3 target now. c29x3.out would have started execution as soon as it is released from reset.
- In case of RAM configuration, restart the CPU3 target and load the symbols.

For FLASH configuration, this example is run in FLASH BANKMODE2, where CPU3 has access to FLASH (FRI-2). Refer to the Flash Plugin documentation to know about changing FLASH BANKMODEs and more. Additionally, the CPAx and CDAx RAMs are used for allocating various RAM sections. *External Connections*
- None.

Watch Variables
- pass

16.6.2.4 IPC basic message passing example with interrupt - MULTI_CORE

FILE: ipc_ex2_basic_cpu1_cpu3_multi_c29x3.c

This example demonstrates how to configure IPC and pass information from C29x1 to C29x3 core without message queues.

When using CCS for debugging this Multi-core example, after launching the debug session,
- Connect to CPU1 and first load the c29x3.out. (When loaded to FLASH, this might not halt at main due to an error, since CPU1 cannot execute from CPU3 FLASH memory (FRI-2)).
- c29x1.out can be loaded next.
- After the program is loaded, run CPU1.
- Once c29x1 configures and releases CPU3 out of reset, the program stops.
- Connect to CPU3 target now. c29x3.out would have started execution as soon as it is released from reset.
- In case of RAM configuration, restart the CPU3 target and load the symbols.
- In case of FLASH, do a Power-on-Reset, connect to CPU1/ CPU3 and load the respective symbols.

For FLASH configuration, this example is run in FLASH BANKMODE2, where CPU3 has access to FLASH (FRI-2). Refer to the Flash Plugin documentation to know about changing FLASH BANKMODEs and more. Additionally, the CPAx and CDAx RAMs are used for allocating various RAM sections. *External Connections*
- None.

Watch Variables
- None.

16.7 IPC Registers

This section describes the IPC Registers.

16.7.1 IPC Base Address Table

Table 16-3. IPC Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
IPC_COUNTER_REGS [1]	IPCCOUNTER_BASE	0x301B_0000	-	YES	YES	YES	-	-	YES	YES
CPU1_IPC_SEND_REGS	CPU1IPCSEND_BASE	0x3022_0000	-	YES	YES	YES	-	-	-	YES
CPU2_IPC_SEND_REGS	CPU2IPCSEND_BASE	0x3022_8000	-	YES	YES	YES	-	-	-	YES
CPU3_IPC_SEND_REGS	CPU3IPCSEND_BASE	0x3023_0000	-	YES	YES	YES	-	-	-	YES
CPU1_IPC_RCV_REGS	CPU1IPCRCV_BASE	0x3024_0000	-	YES	YES	YES	-	-	-	YES
CPU2_IPC_RCV_REGS	CPU2IPCRCV_BASE	0x3024_8000	-	YES	YES	YES	-	-	-	YES
CPU3_IPC_RCV_REGS	CPU3IPCRCV_BASE	0x3025_0000	-	YES	YES	YES	-	-	-	YES

(1) Registers writeable by CPU1.LINK0, CPU1.LINK1, CPU1.LINK2 only. All CPUs can read all registers in all LINKs. Debug write access only allowed if Zone0 or Zone1 are enabled for full debug by all CPUs. Debug reads always allowed. Register Read/Write access by HSM.

16.7.2 IPC_COUNTER_REGS Registers

Table 16-4 lists the memory-mapped registers for the IPC_COUNTER_REGS registers. All register offset addresses not listed in Table 16-4 should be considered as reserved locations and the register contents should not be modified.

Table 16-4. IPC_COUNTER_REGS Registers

Offset	Acronym	Register Name	Protection
0h	IPCCOUNTERL	IPCCOUNTERL Register	
4h	IPCCOUNTERH	IPCCOUNTERH Register	

Complex bit access types are encoded to fit into small table cells. Table 16-5 shows the codes that are used for access types in this section.

Table 16-5. IPC_COUNTER_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Reset or Default Value		
-*n*		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

16.7.2.1 IPCCOUNTERL Register (Offset = 0h) [Reset = 00000000h]

IPCCOUNTERL is shown in Figure 16-2 and described in Table 16-6.

Return to the Summary Table.

IPC Counter High Register

Figure 16-2. IPCCOUNTERL Register

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

Table 16-6. IPCCOUNTERL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNT	R	0h	This is the lower 32-bits of free running 64 bit timestamp counter clocked by the PLLSYSCLK. Reset type: XRSn

16.7.2.2 IPCCOUNTERH Register (Offset = 4h) [Reset = 00000000h]

IPCCOUNTERH is shown in Figure 16-3 and described in Table 16-7.

Return to the Summary Table.

IPC Counter Low Register

Figure 16-3. IPCCOUNTERH Register

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

Table 16-7. IPCCOUNTERH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNT	R	0h	This is the upper 32-bits of free running 64 bit timestamp counter clocked by the PLLSYSCLK. Reset type: XRSn

16.7.3 CPU1_IPC_SEND_REGS Registers

Table 16-8 lists the memory-mapped registers for the CPU1_IPC_SEND_REGS registers. All register offset addresses not listed in Table 16-8 should be considered as reserved locations and the register contents should not be modified.

Table 16-8. CPU1_IPC_SEND_REGS Registers

Offset	Acronym	Register Name	Protection
0h + formula	CPU1TOCPU2INTIPCSET_j	CPU1TOCPU2INTIPCSET Register	
4h + formula	CPU1TOCPU2INTIPCCLR_j	CPU1TOCPU2INTIPCCLR Register	
8h + formula	CPU1TOCPU2INTIPCFLG_j	CPU1TOCPU2INTIPCFLG Register	
10h + formula	CPU1TOCPU2INTIPCSENDCOM_j	CPU1TOCPU2INTIPCSENDCOM Register	
14h + formula	CPU1TOCPU2INTIPCSENDADDR_j	CPU1TOCPU2INTIPCSENDADDR Register	
18h + formula	CPU1TOCPU2INTIPCSENDDATA_j	CPU1TOCPU2INTIPCSENDDATA Register	
1Ch + formula	CPU2TOCPU1INTREMOTEREPLY_j	CPU2TOCPU1INTREMOTEREPLY Register	
2000h + formula	CPU1TOCPU3INTIPCSET_j	CPU1TOCPU3INTIPCSET Register	
2004h + formula	CPU1TOCPU3INTIPCCLR_j	CPU1TOCPU3INTIPCCLR Register	
2008h + formula	CPU1TOCPU3INTIPCFLG_j	CPU1TOCPU3INTIPCFLG Register	
2010h + formula	CPU1TOCPU3INTIPCSENDCOM_j	CPU1TOCPU3INTIPCSENDCOM Register	
2014h + formula	CPU1TOCPU3INTIPCSENDADDR_j	CPU1TOCPU3INTIPCSENDADDR Register	
2018h + formula	CPU1TOCPU3INTIPCSENDDATA_j	CPU1TOCPU3INTIPCSENDDATA Register	
201Ch + formula	CPU3TOCPU1INTREMOTEREPLY_j	CPU3TOCPU1INTREMOTEREPLY Register	
6000h + formula	CPU1TOHSMINTIPCSET_j	CPU1TOHSMINTIPCSET Register	
6004h + formula	CPU1TOHSMINTIPCCLR_j	CPU1TOHSMINTIPCCLR Register	
6008h + formula	CPU1TOHSMINTIPCFLG_j	CPU1TOHSMINTIPCFLG Register	

Complex bit access types are encoded to fit into small table cells. Table 16-9 shows the codes that are used for access types in this section.

Table 16-9. CPU1_IPC_SEND_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		

Table 16-9. CPU1_IPC_SEND_REGS Access Type Codes (continued)

Access Type	Code	Description
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

16.7.3.1 CPU1TOCPU2INTIPCSET_j Register (Offset = 0h + formula) [Reset = 00000000h]

CPU1TOCPU2INTIPCSET_j is shown in Figure 16-4 and described in Table 16-10.

Return to the Summary Table.

Set CPU2TOCPU1IPCFLG register

Offset = 0h + (j * 800h); where j = 0h to 3h

Figure 16-4. CPU1TOCPU2INTIPCSET_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-10. CPU1TOCPU2INTIPCSET_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit sets the IPC31 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit sets the IPC30 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit sets the IPC29 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit sets the IPC28 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit sets the IPC27 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit sets the IPC26 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit sets the IPC25 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
24	IPC24	R-0/W1S	0h	Writing 1 to this bit sets the IPC24 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit sets the IPC23 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn

Table 16-10. CPU1TOCPU2INTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R-0/W1S	0h	Writing 1 to this bit sets the IPC22 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit sets the IPC21 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit sets the IPC20 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit sets the IPC19 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit sets the IPC18 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit sets the IPC17 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit sets the IPC16 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit sets the IPC15 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit sets the IPC14 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit sets the IPC13 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit sets the IPC12 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit sets the IPC11 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
10	IPC10	R-0/W1S	0h	Writing 1 to this bit sets the IPC10 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit sets the IPC9 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit sets the IPC8 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit sets the IPC7 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit sets the IPC6 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn

Table 16-10. CPU1TOCPU2INTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R-0/W1S	0h	Writing 1 to this bit sets the IPC5 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit sets the IPC4 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit sets the IPC3 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit sets the IPC2 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit sets the IPC1 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit sets the IPC0 event flag for the remote CPU. Writing 0 has no effect. Notes: [1] IPC event flags 0 will trigger interrupts in the receiving CPU via the PIPE. Reset type: CPU1.SYSRSn

16.7.3.2 CPU1TOCPU2INTIPCCLR_j Register (Offset = 4h + formula) [Reset = 00000000h]

CPU1TOCPU2INTIPCCLR_j is shown in Figure 16-5 and described in Table 16-11.

Return to the Summary Table.

Clear CPU1TOCPU2IPCFLG register

Offset = 4h + (j * 800h); where j = 0h to 3h

Figure 16-5. CPU1TOCPU2INTIPCCLR_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-11. CPU1TOCPU2INTIPCCLR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC31 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC30 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC29 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC28 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC27 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC26 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC25 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn

Table 16-11. CPU1TOCPU2INTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	IPC24	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC24 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC23 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
22	IPC22	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC22 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC21 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC20 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC19 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC18 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC17 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC16 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC15 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC14 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC13 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC12 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC11 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn

Table 16-11. CPU1TOCPU2INTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	IPC10	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC10 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC9 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC8 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC7 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC6 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
5	IPC5	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC5 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC4 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC3 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC2 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC1 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU2IPCFLG.IPC0 event flag for CPU2. Writing 0 has no effect. Reset type: CPU1.SYSRSn

16.7.3.3 CPU1TOCPU2INTIPCFLG_j Register (Offset = 8h + formula) [Reset = 00000000h]

CPU1TOCPU2INTIPCFLG_j is shown in Figure 16-6 and described in Table 16-12.

Return to the Summary Table.

CPU1TOCPU2INTIPCFLG Register

Offset = 8h + (j * 800h); where j = 0h to 3h

Figure 16-6. CPU1TOCPU2INTIPCFLG_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-12. CPU1TOCPU2INTIPCFLG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R	0h	0: No IPC31 event request to CPU2 1: IPC31 event request to CPU2 Reset type: CPU2.SYSRSn
30	IPC30	R	0h	0: No IPC30 event request to CPU2 1: IPC30 event request to CPU2 Reset type: CPU2.SYSRSn
29	IPC29	R	0h	0: No IPC29 event request to CPU2 1: IPC29 event request to CPU2 Reset type: CPU2.SYSRSn
28	IPC28	R	0h	0: No IPC28 event request to CPU2 1: IPC28 event request to CPU2 Reset type: CPU2.SYSRSn
27	IPC27	R	0h	0: No IPC27 event request to CPU2 1: IPC27 event request to CPU2 Reset type: CPU2.SYSRSn
26	IPC26	R	0h	0: No IPC26 event request to CPU2 1: IPC26 event request to CPU2 Reset type: CPU2.SYSRSn
25	IPC25	R	0h	0: No IPC25 event request to CPU2 1: IPC25 event request to CPU2 Reset type: CPU2.SYSRSn
24	IPC24	R	0h	0: No IPC24 event request to CPU2 1: IPC24 event request to CPU2 Reset type: CPU2.SYSRSn
23	IPC23	R	0h	0: No IPC23 event request to CPU2 1: IPC23 event request to CPU2 Reset type: CPU2.SYSRSn

Table 16-12. CPU1TOCPU2INTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R	0h	0: No IPC22 event request to CPU2 1: IPC22 event request to CPU2 Reset type: CPU2.SYSRSn
21	IPC21	R	0h	0: No IPC21 event request to CPU2 1: IPC21 event request to CPU2 Reset type: CPU2.SYSRSn
20	IPC20	R	0h	0: No IPC20 event request to CPU2 1: IPC20 event request to CPU2 Reset type: CPU2.SYSRSn
19	IPC19	R	0h	0: No IPC19 event request to CPU2 1: IPC19 event request to CPU2 Reset type: CPU2.SYSRSn
18	IPC18	R	0h	0: No IPC18 event request to CPU2 1: IPC18 event request to CPU2 Reset type: CPU2.SYSRSn
17	IPC17	R	0h	0: No IPC17 event request to CPU2 1: IPC17 event request to CPU2 Reset type: CPU2.SYSRSn
16	IPC16	R	0h	0: No IPC16 event request to CPU2 1: IPC16 event request to CPU2 Reset type: CPU2.SYSRSn
15	IPC15	R	0h	0: No IPC15 event request to CPU2 1: IPC15 event request to CPU2 Reset type: CPU2.SYSRSn
14	IPC14	R	0h	0: No IPC14 event request to CPU2 1: IPC14 event request to CPU2 Reset type: CPU2.SYSRSn
13	IPC13	R	0h	0: No IPC13 event request to CPU2 1: IPC13 event request to CPU2 Reset type: CPU2.SYSRSn
12	IPC12	R	0h	0: No IPC12 event request to CPU2 1: IPC12 event request to CPU2 Reset type: CPU2.SYSRSn
11	IPC11	R	0h	0: No IPC11 event request to CPU2 1: IPC11 event request to CPU2 Reset type: CPU2.SYSRSn
10	IPC10	R	0h	0: No IPC10 event request to CPU2 1: IPC10 event request to CPU2 Reset type: CPU2.SYSRSn
9	IPC9	R	0h	0: No IPC9 event request to CPU2 1: IPC9 event request to CPU2 Reset type: CPU2.SYSRSn
8	IPC8	R	0h	0: No IPC8 event request to CPU2 1: IPC8 event request to CPU2 Reset type: CPU2.SYSRSn
7	IPC7	R	0h	0: No IPC7 event request to CPU2 1: IPC7 event request to CPU2 Reset type: CPU2.SYSRSn
6	IPC6	R	0h	0: No IPC6 event request to CPU2 1: IPC6 event request to CPU2 Reset type: CPU2.SYSRSn

Table 16-12. CPU1TOCPU2INTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R	0h	0: No IPC5 event request to CPU2 1: IPC5 event request to CPU2 Reset type: CPU2.SYSRSn
4	IPC4	R	0h	0: No IPC4 event request to CPU2 1: IPC4 event request to CPU2 Reset type: CPU2.SYSRSn
3	IPC3	R	0h	0: No IPC3 event request to CPU2 1: IPC3 event request to CPU2 Reset type: CPU2.SYSRSn
2	IPC2	R	0h	0: No IPC2 event request to CPU2 1: IPC2 event request to CPU2 Reset type: CPU2.SYSRSn
1	IPC1	R	0h	0: No IPC1 event request to CPU2 1: IPC1 event request to CPU2 Reset type: CPU2.SYSRSn
0	IPC0	R	0h	0: No IPC0 event request to CPU2 1: IPC0 event request to CPU2 Reset type: CPU2.SYSRSn

16.7.3.4 CPU1TOCPU2INTIPCSENDCOM_j Register (Offset = 10h + formula) [Reset = 00000000h]

CPU1TOCPU2INTIPCSENDCOM_j is shown in Figure 16-7 and described in Table 16-13.

Return to the Summary Table.

CPU1 to CPU2 IPC Command

Offset = 10h + (j * 800h); where j = 0h to 3h

Figure 16-7. CPU1TOCPU2INTIPCSENDCOM_j Register

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

Table 16-13. CPU1TOCPU2INTIPCSENDCOM_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMMAND	R/W	0h	This is a general purpose register used to send software-defined commands to from CPU1 to CPU2 Reset type: CPU1.SYSRSn

16.7.3.5 CPU1TOCPU2INTIPCSENDADDR_j Register (Offset = 14h + formula) [Reset = 00000000h]

CPU1TOCPU2INTIPCSENDADDR_j is shown in Figure 16-8 and described in Table 16-14.

Return to the Summary Table.

CPU1 to CPU2 IPC Address

Offset = 14h + (j * 800h); where j = 0h to 3h

Figure 16-8. CPU1TOCPU2INTIPCSENDADDR_j Register

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

Table 16-14. CPU1TOCPU2INTIPCSENDADDR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R/W	0h	This is a general purpose register used to send software-defined Address to from CPU1 to CPU2 Reset type: CPU1.SYSRSn

16.7.3.6 CPU1TOCPU2INTIPCSENDDATA_j Register (Offset = 18h + formula) [Reset = 00000000h]

CPU1TOCPU2INTIPCSENDDATA_j is shown in Figure 16-9 and described in Table 16-15.

Return to the Summary Table.

CPU1 to CPU2 IPC Data

Offset = 18h + (j * 800h); where j = 0h to 3h

Figure 16-9. CPU1TOCPU2INTIPCSENDDATA_j Register

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

Table 16-15. CPU1TOCPU2INTIPCSENDDATA_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/W	0h	This is a general purpose register used to send software-defined Data to from CPU1 to CPU2 Reset type: CPU1.SYSRSn

16.7.3.7 CPU2TOCPU1INTREMOTEREPLY_j Register (Offset = 1Ch + formula) [Reset = 00000000h]

CPU2TOCPU1INTREMOTEREPLY_j is shown in Figure 16-10 and described in Table 16-16.

Return to the Summary Table.

Reply from CPU2 to CPU1TOCPU2IPCSENDCOM command request

Offset = 1Ch + (j * 800h); where j = 0h to 3h

Figure 16-10. CPU2TOCPU1INTREMOTEREPLY_j Register

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

Table 16-16. CPU2TOCPU1INTREMOTEREPLY_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REPLY	R	0h	Refelects the state of CPU2TOCPU1INT IPCREPLY register Reset type: CPU1.SYSRSn

16.7.3.8 CPU1TOCPU3INTIPCSET_j Register (Offset = 2000h + formula) [Reset = 00000000h]

CPU1TOCPU3INTIPCSET_j is shown in Figure 16-11 and described in Table 16-17.

Return to the Summary Table.

Set CPU1TOCPU3INTIPCSET Register

Offset = 2000h + (j * 800h); where j = 0h to 3h

Figure 16-11. CPU1TOCPU3INTIPCSET_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-17. CPU1TOCPU3INTIPCSET_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit sets the IPC31 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit sets the IPC30 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit sets the IPC29 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit sets the IPC28 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit sets the IPC27 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit sets the IPC26 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit sets the IPC25 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
24	IPC24	R-0/W1S	0h	Writing 1 to this bit sets the IPC24 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit sets the IPC23 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn

Table 16-17. CPU1TOCPU3INTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R-0/W1S	0h	Writing 1 to this bit sets the IPC22 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit sets the IPC21 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit sets the IPC20 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit sets the IPC19 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit sets the IPC18 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit sets the IPC17 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit sets the IPC16 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit sets the IPC15 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit sets the IPC14 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit sets the IPC13 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit sets the IPC12 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit sets the IPC11 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
10	IPC10	R-0/W1S	0h	Writing 1 to this bit sets the IPC10 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit sets the IPC9 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit sets the IPC8 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit sets the IPC7 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit sets the IPC6 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn

Table 16-17. CPU1TOCPU3INTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R-0/W1S	0h	Writing 1 to this bit sets the IPC5 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit sets the IPC4 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit sets the IPC3 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit sets the IPC2 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit sets the IPC1 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit sets the IPC0 event flag for the remote CPU. Writing 0 has no effect. Notes: [1] IPC event flags 0 will trigger interrupts in the receiving CPU via the PIPE. Reset type: CPU1.SYSRSn

16.7.3.9 CPU1TOCPU3INTIPCCLR_j Register (Offset = 2004h + formula) [Reset = 00000000h]

CPU1TOCPU3INTIPCCLR_j is shown in Figure 16-12 and described in Table 16-18.

Return to the Summary Table.

Clear CPU1TOCPU3IPCFLG register

Offset = 2004h + (j * 800h); where j = 0h to 3h

Figure 16-12. CPU1TOCPU3INTIPCCLR_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-18. CPU1TOCPU3INTIPCCLR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC31 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC30 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC29 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC28 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC27 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC26 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC25 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn

Table 16-18. CPU1TOCPU3INTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	IPC24	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC24 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC23 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
22	IPC22	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC22 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC21 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC20 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC19 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC18 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC17 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC16 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC15 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC14 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC13 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC12 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC11 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn

Table 16-18. CPU1TOCPU3INTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	IPC10	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC10 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC9 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC8 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC7 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC6 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
5	IPC5	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC5 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC4 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC3 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC2 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC1 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOCPU3IPCFLG.IPC0 event flag for CPU3. Writing 0 has no effect. Reset type: CPU1.SYSRSn

16.7.3.10 CPU1TOCPU3INTIPCFLG_j Register (Offset = 2008h + formula) [Reset = 00000000h]

CPU1TOCPU3INTIPCFLG_j is shown in Figure 16-13 and described in Table 16-19.

Return to the Summary Table.

CPU1TOCPU3INTIPCFLG Register

Offset = 2008h + (j * 800h); where j = 0h to 3h

Figure 16-13. CPU1TOCPU3INTIPCFLG_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-19. CPU1TOCPU3INTIPCFLG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R	0h	0: No IPC31 event request to CPU3 1: IPC31 event request to CPU3 Reset type: CPUx.SYSRSn
30	IPC30	R	0h	0: No IPC30 event request to CPU3 1: IPC30 event request to CPU3 Reset type: CPUx.SYSRSn
29	IPC29	R	0h	0: No IPC29 event request to CPU3 1: IPC29 event request to CPU3 Reset type: CPUx.SYSRSn
28	IPC28	R	0h	0: No IPC28 event request to CPU3 1: IPC28 event request to CPU3 Reset type: CPUx.SYSRSn
27	IPC27	R	0h	0: No IPC27 event request to CPU3 1: IPC27 event request to CPU3 Reset type: CPUx.SYSRSn
26	IPC26	R	0h	0: No IPC26 event request to CPU3 1: IPC26 event request to CPU3 Reset type: CPUx.SYSRSn
25	IPC25	R	0h	0: No IPC25 event request to CPU3 1: IPC25 event request to CPU3 Reset type: CPUx.SYSRSn
24	IPC24	R	0h	0: No IPC24 event request to CPU3 1: IPC24 event request to CPU3 Reset type: CPUx.SYSRSn
23	IPC23	R	0h	0: No IPC23 event request to CPU3 1: IPC23 event request to CPU3 Reset type: CPUx.SYSRSn

Table 16-19. CPU1TOCPU3INTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R	0h	0: No IPC22 event request to CPU3 1: IPC22 event request to CPU3 Reset type: CPUx.SYSRSn
21	IPC21	R	0h	0: No IPC21 event request to CPU3 1: IPC21 event request to CPU3 Reset type: CPUx.SYSRSn
20	IPC20	R	0h	0: No IPC20 event request to CPU3 1: IPC20 event request to CPU3 Reset type: CPUx.SYSRSn
19	IPC19	R	0h	0: No IPC19 event request to CPU3 1: IPC19 event request to CPU3 Reset type: CPUx.SYSRSn
18	IPC18	R	0h	0: No IPC18 event request to CPU3 1: IPC18 event request to CPU3 Reset type: CPUx.SYSRSn
17	IPC17	R	0h	0: No IPC17 event request to CPU3 1: IPC17 event request to CPU3 Reset type: CPUx.SYSRSn
16	IPC16	R	0h	0: No IPC16 event request to CPU3 1: IPC16 event request to CPU3 Reset type: CPUx.SYSRSn
15	IPC15	R	0h	0: No IPC15 event request to CPU3 1: IPC15 event request to CPU3 Reset type: CPUx.SYSRSn
14	IPC14	R	0h	0: No IPC14 event request to CPU3 1: IPC14 event request to CPU3 Reset type: CPUx.SYSRSn
13	IPC13	R	0h	0: No IPC13 event request to CPU3 1: IPC13 event request to CPU3 Reset type: CPUx.SYSRSn
12	IPC12	R	0h	0: No IPC12 event request to CPU3 1: IPC12 event request to CPU3 Reset type: CPUx.SYSRSn
11	IPC11	R	0h	0: No IPC11 event request to CPU3 1: IPC11 event request to CPU3 Reset type: CPUx.SYSRSn
10	IPC10	R	0h	0: No IPC10 event request to CPU3 1: IPC10 event request to CPU3 Reset type: CPUx.SYSRSn
9	IPC9	R	0h	0: No IPC9 event request to CPU3 1: IPC9 event request to CPU3 Reset type: CPUx.SYSRSn
8	IPC8	R	0h	0: No IPC8 event request to CPU3 1: IPC8 event request to CPU3 Reset type: CPUx.SYSRSn
7	IPC7	R	0h	0: No IPC7 event request to CPU3 1: IPC7 event request to CPU3 Reset type: CPUx.SYSRSn
6	IPC6	R	0h	0: No IPC6 event request to CPU3 1: IPC6 event request to CPU3 Reset type: CPUx.SYSRSn

Table 16-19. CPU1TOCPU3INTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R	0h	0: No IPC5 event request to CPU3 1: IPC5 event request to CPU3 Reset type: CPUx.SYSRSn
4	IPC4	R	0h	0: No IPC4 event request to CPU3 1: IPC4 event request to CPU3 Reset type: CPUx.SYSRSn
3	IPC3	R	0h	0: No IPC3 event request to CPU3 1: IPC3 event request to CPU3 Reset type: CPUx.SYSRSn
2	IPC2	R	0h	0: No IPC2 event request to CPU3 1: IPC2 event request to CPU3 Reset type: CPUx.SYSRSn
1	IPC1	R	0h	0: No IPC1 event request to CPU3 1: IPC1 event request to CPU3 Reset type: CPUx.SYSRSn
0	IPC0	R	0h	0: No IPC0 event request to CPU3 1: IPC0 event request to CPU3 Reset type: CPUx.SYSRSn

16.7.3.11 CPU1TOCPU3INTIPCSENDCOM_j Register (Offset = 2010h + formula) [Reset = 00000000h]

CPU1TOCPU3INTIPCSENDCOM_j is shown in Figure 16-14 and described in Table 16-20.

Return to the Summary Table.

CPU1 to CPU3 IPC Command

Offset = 2010h + (j * 800h); where j = 0h to 3h

Figure 16-14. CPU1TOCPU3INTIPCSENDCOM_j Register

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

Table 16-20. CPU1TOCPU3INTIPCSENDCOM_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMMAND	R/W	0h	This is a general purpose register used to send software-defined commands to from CPU1 to CPU3 Reset type: CPU1.SYSRSn

16.7.3.12 CPU1TOCPU3INTIPCSENDADDR_j Register (Offset = 2014h + formula) [Reset = 00000000h]

CPU1TOCPU3INTIPCSENDADDR_j is shown in Figure 16-15 and described in Table 16-21.

Return to the Summary Table.

CPU1 to CPU3 IPC Address

Offset = 2014h + (j * 800h); where j = 0h to 3h

Figure 16-15. CPU1TOCPU3INTIPCSENDADDR_j Register

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

Table 16-21. CPU1TOCPU3INTIPCSENDADDR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R/W	0h	This is a general purpose register used to send software-defined Address to from CPU1 to CPU3 Reset type: CPU1.SYSRSn

16.7.3.13 CPU1TOCPU3INTIPCSENDDATA_j Register (Offset = 2018h + formula) [Reset = 00000000h]

CPU1TOCPU3INTIPCSENDDATA_j is shown in Figure 16-16 and described in Table 16-22.

Return to the Summary Table.

CPU1 to CPU3 IPC Data

Offset = 2018h + (j * 800h); where j = 0h to 3h

Figure 16-16. CPU1TOCPU3INTIPCSENDDATA_j Register

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

Table 16-22. CPU1TOCPU3INTIPCSENDDATA_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/W	0h	This is a general purpose register used to send software-defined Data to from CPU1 to CPU3 Reset type: CPU1.SYSRSn

16.7.3.14 CPU3TOCPU1INTREMOTEREPLY_j Register (Offset = 201Ch + formula) [Reset = 00000000h]

CPU3TOCPU1INTREMOTEREPLY_j is shown in Figure 16-17 and described in Table 16-23.

Return to the Summary Table.

Reply from CPU3 to CPU1TOCPU3IPCSENDCOM command request

Offset = 201Ch + (j * 800h); where j = 0h to 3h

Figure 16-17. CPU3TOCPU1INTREMOTEREPLY_j Register

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

Table 16-23. CPU3TOCPU1INTREMOTEREPLY_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REPLY	R	0h	Refelects the state of CPU3TOCPU1INT IPCREPLY register Reset type: CPU1.SYSRSn

16.7.3.15 CPU1TOHSMINTIPCSET_j Register (Offset = 6000h + formula) [Reset = 00000000h]

CPU1TOHSMINTIPCSET_j is shown in Figure 16-18 and described in Table 16-24.

Return to the Summary Table.

Set CPU1TOHSMINTIPCSET Register

Offset = 6000h + (j * 1000h); where j = 0h to 1h

Figure 16-18. CPU1TOHSMINTIPCSET_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-24. CPU1TOHSMINTIPCSET_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit sets the IPC31 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit sets the IPC30 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit sets the IPC29 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit sets the IPC28 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit sets the IPC27 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit sets the IPC26 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit sets the IPC25 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
24	IPC24	R-0/W1S	0h	Writing 1 to this bit sets the IPC24 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit sets the IPC23 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn

Table 16-24. CPU1TOHSMINTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R-0/W1S	0h	Writing 1 to this bit sets the IPC22 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit sets the IPC21 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit sets the IPC20 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit sets the IPC19 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit sets the IPC18 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit sets the IPC17 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit sets the IPC16 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit sets the IPC15 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit sets the IPC14 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit sets the IPC13 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit sets the IPC12 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit sets the IPC11 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
10	IPC10	R-0/W1S	0h	Writing 1 to this bit sets the IPC10 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit sets the IPC9 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit sets the IPC8 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit sets the IPC7 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit sets the IPC6 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn

Table 16-24. CPU1TOHSMINTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R-0/W1S	0h	Writing 1 to this bit sets the IPC5 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit sets the IPC4 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit sets the IPC3 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit sets the IPC2 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit sets the IPC1 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU1.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit sets the IPC0 event flag for the remote CPU. Writing 0 has no effect. Notes: [1] IPC event flags 0 will trigger interrupts in the receiving CPU via the PIPE. Reset type: CPU1.SYSRSn

16.7.3.16 CPU1TOHSMINTIPCCLR_j Register (Offset = 6004h + formula) [Reset = 00000000h]

CPU1TOHSMINTIPCCLR_j is shown in Figure 16-19 and described in Table 16-25.

Return to the Summary Table.

Clear CPU1TOHSMIPCFLG register

Offset = 6004h + (j * 1000h); where j = 0h to 1h

Figure 16-19. CPU1TOHSMINTIPCCLR_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-25. CPU1TOHSMINTIPCCLR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC31 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC30 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC29 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC28 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC27 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC26 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC25 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn

Table 16-25. CPU1TOHSMINTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	IPC24	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC24 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC23 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
22	IPC22	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC22 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC21 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC20 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC19 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC18 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC17 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC16 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC15 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC14 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC13 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC12 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC11 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn

Table 16-25. CPU1TOHSMINTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	IPC10	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC10 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC9 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC8 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC7 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC6 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
5	IPC5	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC5 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC4 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC3 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC2 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC1 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit clear the CPU1TOHSMIPCFLG.IPC0 event flag for HSM. Writing 0 has no effect. Reset type: CPU1.SYSRSn

16.7.3.17 CPU1TOHSMINTIPCFLG_j Register (Offset = 6008h + formula) [Reset = 00000000h]

CPU1TOHSMINTIPCFLG_j is shown in Figure 16-20 and described in Table 16-26.

Return to the Summary Table.

CPU1TOHSMINTIPCFLG Register

Offset = 6008h + (j * 1000h); where j = 0h to 1h

Figure 16-20. CPU1TOHSMINTIPCFLG_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-26. CPU1TOHSMINTIPCFLG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R	0h	0: No IPC31 event request to CPU1 1: IPC31 event request to CPU1 Reset type: CPU1.SYSRSn
30	IPC30	R	0h	0: No IPC30 event request to CPU1 1: IPC30 event request to CPU1 Reset type: CPU1.SYSRSn
29	IPC29	R	0h	0: No IPC29 event request to CPU1 1: IPC29 event request to CPU1 Reset type: CPU1.SYSRSn
28	IPC28	R	0h	0: No IPC28 event request to CPU1 1: IPC28 event request to CPU1 Reset type: CPU1.SYSRSn
27	IPC27	R	0h	0: No IPC27 event request to CPU1 1: IPC27 event request to CPU1 Reset type: CPU1.SYSRSn
26	IPC26	R	0h	0: No IPC26 event request to CPU1 1: IPC26 event request to CPU1 Reset type: CPU1.SYSRSn
25	IPC25	R	0h	0: No IPC25 event request to CPU1 1: IPC25 event request to CPU1 Reset type: CPU1.SYSRSn
24	IPC24	R	0h	0: No IPC24 event request to CPU1 1: IPC24 event request to CPU1 Reset type: CPU1.SYSRSn
23	IPC23	R	0h	0: No IPC23 event request to CPU1 1: IPC23 event request to CPU1 Reset type: CPU1.SYSRSn

Table 16-26. CPU1TOHSMINTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R	0h	0: No IPC22 event request to CPU1 1: IPC22 event request to CPU1 Reset type: CPU1.SYSRSn
21	IPC21	R	0h	0: No IPC21 event request to CPU1 1: IPC21 event request to CPU1 Reset type: CPU1.SYSRSn
20	IPC20	R	0h	0: No IPC20 event request to CPU1 1: IPC20 event request to CPU1 Reset type: CPU1.SYSRSn
19	IPC19	R	0h	0: No IPC19 event request to CPU1 1: IPC19 event request to CPU1 Reset type: CPU1.SYSRSn
18	IPC18	R	0h	0: No IPC18 event request to CPU1 1: IPC18 event request to CPU1 Reset type: CPU1.SYSRSn
17	IPC17	R	0h	0: No IPC17 event request to CPU1 1: IPC17 event request to CPU1 Reset type: CPU1.SYSRSn
16	IPC16	R	0h	0: No IPC16 event request to CPU1 1: IPC16 event request to CPU1 Reset type: CPU1.SYSRSn
15	IPC15	R	0h	0: No IPC15 event request to CPU1 1: IPC15 event request to CPU1 Reset type: CPU1.SYSRSn
14	IPC14	R	0h	0: No IPC14 event request to CPU1 1: IPC14 event request to CPU1 Reset type: CPU1.SYSRSn
13	IPC13	R	0h	0: No IPC13 event request to CPU1 1: IPC13 event request to CPU1 Reset type: CPU1.SYSRSn
12	IPC12	R	0h	0: No IPC12 event request to CPU1 1: IPC12 event request to CPU1 Reset type: CPU1.SYSRSn
11	IPC11	R	0h	0: No IPC11 event request to CPU1 1: IPC11 event request to CPU1 Reset type: CPU1.SYSRSn
10	IPC10	R	0h	0: No IPC10 event request to CPU1 1: IPC10 event request to CPU1 Reset type: CPU1.SYSRSn
9	IPC9	R	0h	0: No IPC9 event request to CPU1 1: IPC9 event request to CPU1 Reset type: CPU1.SYSRSn
8	IPC8	R	0h	0: No IPC8 event request to CPU1 1: IPC8 event request to CPU1 Reset type: CPU1.SYSRSn
7	IPC7	R	0h	0: No IPC7 event request to CPU1 1: IPC7 event request to CPU1 Reset type: CPU1.SYSRSn
6	IPC6	R	0h	0: No IPC6 event request to CPU1 1: IPC6 event request to CPU1 Reset type: CPU1.SYSRSn

Table 16-26. CPU1TOHSMINTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R	0h	0: No IPC5 event request to CPU1 1: IPC5 event request to CPU1 Reset type: CPU1.SYSRSn
4	IPC4	R	0h	0: No IPC4 event request to CPU1 1: IPC4 event request to CPU1 Reset type: CPU1.SYSRSn
3	IPC3	R	0h	0: No IPC3 event request to CPU1 1: IPC3 event request to CPU1 Reset type: CPU1.SYSRSn
2	IPC2	R	0h	0: No IPC2 event request to CPU1 1: IPC2 event request to CPU1 Reset type: CPU1.SYSRSn
1	IPC1	R	0h	0: No IPC1 event request to CPU1 1: IPC1 event request to CPU1 Reset type: CPU1.SYSRSn
0	IPC0	R	0h	0: No IPC0 event request to CPU1 1: IPC0 event request to CPU1 Reset type: CPU1.SYSRSn

16.7.4 CPU2_IPC_SEND_REGS Registers

Table 16-27 lists the memory-mapped registers for the CPU2_IPC_SEND_REGS registers. All register offset addresses not listed in Table 16-27 should be considered as reserved locations and the register contents should not be modified.

Table 16-27. CPU2_IPC_SEND_REGS Registers

Offset	Acronym	Register Name	Protection
0h + formula	CPU2TOCPU1INTIPCSET_j	CPU2TOCPU1INTIPCSET Register	
4h + formula	CPU2TOCPU1INTIPCCLR_j	CPU2TOCPU1INTIPCCLR Register	
8h + formula	CPU2TOCPU1INTIPCFLG_j	CPU2TOCPU1INTIPCFLG Register	
10h + formula	CPU2TOCPU1INTIPCSENDCOM_j	CPU2TOCPU1INTIPCSENDCOM Register	
14h + formula	CPU2TOCPU1INTIPCSENDADDR_j	CPU2TOCPU1INTIPCSENDADDR Register	
18h + formula	CPU2TOCPU1INTIPCSENDDATA_j	CPU2TOCPU1INTIPCSENDDATA Register	
1Ch + formula	CPU1TOCPU2INTREMOTEREPLY_j	CPU1TOCPU2INTREMOTEREPLY Register	
2000h + formula	CPU2TOCPU3INTIPCSET_j	CPU2TOCPU3INTIPCSET Register	
2004h + formula	CPU2TOCPU3INTIPCCLR_j	CPU2TOCPU3INTIPCCLR Register	
2008h + formula	CPU2TOCPU3INTIPCFLG_j	CPU2TOCPU3INTIPCFLG Register	
2010h + formula	CPU2TOCPU3INTIPCSENDCOM_j	CPU2TOCPU3INTIPCSENDCOM Register	
2014h + formula	CPU2TOCPU3INTIPCSENDADDR_j	CPU2TOCPU3INTIPCSENDADDR Register	
2018h + formula	CPU2TOCPU3INTIPCSENDDATA_j	CPU2TOCPU3INTIPCSENDDATA Register	
201Ch + formula	CPU3TOCPU2INTREMOTEREPLY_j	CPU3TOCPU2INTREMOTEREPLY Register	
6000h + formula	CPU2TOHSMINTIPCSET_j	CPU2TOHSMINTIPCSET Register	
6004h + formula	CPU2TOHSMINTIPCCLR_j	CPU2TOHSMINTIPCCLR Register	
6008h + formula	CPU2TOHSMINTIPCFLG_j	CPU2TOHSMINTIPCFLG Register	

Complex bit access types are encoded to fit into small table cells. Table 16-28 shows the codes that are used for access types in this section.

Table 16-28. CPU2_IPC_SEND_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		

Table 16-28. CPU2_IPC_SEND_REGS Access Type Codes (continued)

Access Type	Code	Description
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

16.7.4.1 CPU2TOCPU1INTIPCSET_j Register (Offset = 0h + formula) [Reset = 00000000h]

CPU2TOCPU1INTIPCSET_j is shown in Figure 16-21 and described in Table 16-29.

Return to the Summary Table.

Set CPU1TOCPU2IPCFLG register

Offset = 0h + (j * 800h); where j = 0h to 3h

Figure 16-21. CPU2TOCPU1INTIPCSET_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-29. CPU2TOCPU1INTIPCSET_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit sets the IPC31 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit sets the IPC30 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit sets the IPC29 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit sets the IPC28 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit sets the IPC27 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit sets the IPC26 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit sets the IPC25 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
24	IPC24	R-0/W1S	0h	Writing 1 to this bit sets the IPC24 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit sets the IPC23 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn

Table 16-29. CPU2TOCPU1INTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R-0/W1S	0h	Writing 1 to this bit sets the IPC22 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit sets the IPC21 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit sets the IPC20 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit sets the IPC19 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit sets the IPC18 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit sets the IPC17 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit sets the IPC16 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit sets the IPC15 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit sets the IPC14 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit sets the IPC13 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit sets the IPC12 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit sets the IPC11 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
10	IPC10	R-0/W1S	0h	Writing 1 to this bit sets the IPC10 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit sets the IPC9 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit sets the IPC8 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit sets the IPC7 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit sets the IPC6 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn

Table 16-29. CPU2TOCPU1INTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R-0/W1S	0h	Writing 1 to this bit sets the IPC5 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit sets the IPC4 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit sets the IPC3 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit sets the IPC2 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit sets the IPC1 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit sets the IPC0 event flag for the remote CPU. Writing 0 has no effect. Notes: [1] IPC event flags 0 will trigger interrupts in the receiving CPU via the PIPE. Reset type: CPU2.SYSRSn

16.7.4.2 CPU2TOCPU1INTIPCCLR_j Register (Offset = 4h + formula) [Reset = 00000000h]

CPU2TOCPU1INTIPCCLR_j is shown in Figure 16-22 and described in Table 16-30.

Return to the Summary Table.

Clear CPU2TOCPU1IPCFLG register

Offset = 4h + (j * 800h); where j = 0h to 3h

Figure 16-22. CPU2TOCPU1INTIPCCLR_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-30. CPU2TOCPU1INTIPCCLR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC31 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC30 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC29 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC28 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC27 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC26 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC25 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn

Table 16-30. CPU2TOCPU1INTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	IPC24	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC24 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC23 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
22	IPC22	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC22 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC21 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC20 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC19 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC18 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC17 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC16 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC15 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC14 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC13 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC12 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC11 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn

Table 16-30. CPU2TOCPU1INTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	IPC10	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC10 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC9 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC8 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC7 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC6 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
5	IPC5	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC5 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC4 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC3 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC2 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC1 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU1IPCFLG.IPC0 event flag for CPU1. Writing 0 has no effect. Reset type: CPU2.SYSRSn

16.7.4.3 CPU2TOCPU1INTIPCFLG_j Register (Offset = 8h + formula) [Reset = 00000000h]

CPU2TOCPU1INTIPCFLG_j is shown in Figure 16-23 and described in Table 16-31.

Return to the Summary Table.

CPU2TOCPU1INTIPCFLG Register

Offset = 8h + (j * 800h); where j = 0h to 3h

Figure 16-23. CPU2TOCPU1INTIPCFLG_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-31. CPU2TOCPU1INTIPCFLG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R	0h	0: No IPC31 event request to CPU1 1: IPC31 event request to CPU1 Reset type: CPU1.SYSRSn
30	IPC30	R	0h	0: No IPC30 event request to CPU1 1: IPC30 event request to CPU1 Reset type: CPU1.SYSRSn
29	IPC29	R	0h	0: No IPC29 event request to CPU1 1: IPC29 event request to CPU1 Reset type: CPU1.SYSRSn
28	IPC28	R	0h	0: No IPC28 event request to CPU1 1: IPC28 event request to CPU1 Reset type: CPU1.SYSRSn
27	IPC27	R	0h	0: No IPC27 event request to CPU1 1: IPC27 event request to CPU1 Reset type: CPU1.SYSRSn
26	IPC26	R	0h	0: No IPC26 event request to CPU1 1: IPC26 event request to CPU1 Reset type: CPU1.SYSRSn
25	IPC25	R	0h	0: No IPC25 event request to CPU1 1: IPC25 event request to CPU1 Reset type: CPU1.SYSRSn
24	IPC24	R	0h	0: No IPC24 event request to CPU1 1: IPC24 event request to CPU1 Reset type: CPU1.SYSRSn
23	IPC23	R	0h	0: No IPC23 event request to CPU1 1: IPC23 event request to CPU1 Reset type: CPU1.SYSRSn

Table 16-31. CPU2TOCPU1INTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R	0h	0: No IPC22 event request to CPU1 1: IPC22 event request to CPU1 Reset type: CPU1.SYSRSn
21	IPC21	R	0h	0: No IPC21 event request to CPU1 1: IPC21 event request to CPU1 Reset type: CPU1.SYSRSn
20	IPC20	R	0h	0: No IPC20 event request to CPU1 1: IPC20 event request to CPU1 Reset type: CPU1.SYSRSn
19	IPC19	R	0h	0: No IPC19 event request to CPU1 1: IPC19 event request to CPU1 Reset type: CPU1.SYSRSn
18	IPC18	R	0h	0: No IPC18 event request to CPU1 1: IPC18 event request to CPU1 Reset type: CPU1.SYSRSn
17	IPC17	R	0h	0: No IPC17 event request to CPU1 1: IPC17 event request to CPU1 Reset type: CPU1.SYSRSn
16	IPC16	R	0h	0: No IPC16 event request to CPU1 1: IPC16 event request to CPU1 Reset type: CPU1.SYSRSn
15	IPC15	R	0h	0: No IPC15 event request to CPU1 1: IPC15 event request to CPU1 Reset type: CPU1.SYSRSn
14	IPC14	R	0h	0: No IPC14 event request to CPU1 1: IPC14 event request to CPU1 Reset type: CPU1.SYSRSn
13	IPC13	R	0h	0: No IPC13 event request to CPU1 1: IPC13 event request to CPU1 Reset type: CPU1.SYSRSn
12	IPC12	R	0h	0: No IPC12 event request to CPU1 1: IPC12 event request to CPU1 Reset type: CPU1.SYSRSn
11	IPC11	R	0h	0: No IPC11 event request to CPU1 1: IPC11 event request to CPU1 Reset type: CPU1.SYSRSn
10	IPC10	R	0h	0: No IPC10 event request to CPU1 1: IPC10 event request to CPU1 Reset type: CPU1.SYSRSn
9	IPC9	R	0h	0: No IPC9 event request to CPU1 1: IPC9 event request to CPU1 Reset type: CPU1.SYSRSn
8	IPC8	R	0h	0: No IPC8 event request to CPU1 1: IPC8 event request to CPU1 Reset type: CPU1.SYSRSn
7	IPC7	R	0h	0: No IPC7 event request to CPU1 1: IPC7 event request to CPU1 Reset type: CPU1.SYSRSn
6	IPC6	R	0h	0: No IPC6 event request to CPU1 1: IPC6 event request to CPU1 Reset type: CPU1.SYSRSn

Table 16-31. CPU2TOCPU1INTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R	0h	0: No IPC5 event request to CPU1 1: IPC5 event request to CPU1 Reset type: CPU1.SYSRSn
4	IPC4	R	0h	0: No IPC4 event request to CPU1 1: IPC4 event request to CPU1 Reset type: CPU1.SYSRSn
3	IPC3	R	0h	0: No IPC3 event request to CPU1 1: IPC3 event request to CPU1 Reset type: CPU1.SYSRSn
2	IPC2	R	0h	0: No IPC2 event request to CPU1 1: IPC2 event request to CPU1 Reset type: CPU1.SYSRSn
1	IPC1	R	0h	0: No IPC1 event request to CPU1 1: IPC1 event request to CPU1 Reset type: CPU1.SYSRSn
0	IPC0	R	0h	0: No IPC0 event request to CPU1 1: IPC0 event request to CPU1 Reset type: CPU1.SYSRSn

16.7.4.4 CPU2TOCPU1INTIPCSENDCOM_j Register (Offset = 10h + formula) [Reset = 00000000h]

CPU2TOCPU1INTIPCSENDCOM_j is shown in Figure 16-24 and described in Table 16-32.

Return to the Summary Table.

CPU2 to CPU1 IPC Command

Offset = 10h + (j * 800h); where j = 0h to 3h

Figure 16-24. CPU2TOCPU1INTIPCSENDCOM_j Register

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

Table 16-32. CPU2TOCPU1INTIPCSENDCOM_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMMAND	R/W	0h	This is a general purpose register used to send software-defined commands to from CPU2 to CPU1 Reset type: CPU2.SYSRSn

16.7.4.5 CPU2TOCPU1INTIPCSENDADDR_j Register (Offset = 14h + formula) [Reset = 00000000h]

CPU2TOCPU1INTIPCSENDADDR_j is shown in Figure 16-25 and described in Table 16-33.

Return to the Summary Table.

CPU2 to CPU1 IPC Address

Offset = 14h + (j * 800h); where j = 0h to 3h

Figure 16-25. CPU2TOCPU1INTIPCSENDADDR_j Register

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

Table 16-33. CPU2TOCPU1INTIPCSENDADDR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R/W	0h	This is a general purpose register used to send software-defined Address to from CPU2 to CPU1 Reset type: CPU2.SYSRSn

16.7.4.6 CPU2TOCPU1INTIPCSENDDATA_j Register (Offset = 18h + formula) [Reset = 00000000h]

CPU2TOCPU1INTIPCSENDDATA_j is shown in Figure 16-26 and described in Table 16-34.

Return to the Summary Table.

CPU2 to CPU1 IPC Data

Offset = 18h + (j * 800h); where j = 0h to 3h

Figure 16-26. CPU2TOCPU1INTIPCSENDDATA_j Register

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

Table 16-34. CPU2TOCPU1INTIPCSENDDATA_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/W	0h	This is a general purpose register used to send software-defined Data to from CPU2 to CPU1 Reset type: CPU2.SYSRSn

16.7.4.7 CPU1TOCPU2INTREMOTEREPLY_j Register (Offset = 1Ch + formula) [Reset = 00000000h]

CPU1TOCPU2INTREMOTEREPLY_j is shown in Figure 16-27 and described in Table 16-35.

Return to the Summary Table.

Reply from CPU1 to CPU2TOCPU1IPCSENDCOM command request

Offset = 1Ch + (j * 800h); where j = 0h to 3h

Figure 16-27. CPU1TOCPU2INTREMOTEREPLY_j Register

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

Table 16-35. CPU1TOCPU2INTREMOTEREPLY_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REPLY	R	0h	Refelects the state of CPU1TOCPU2INT IPCREPLY register Reset type: CPU2.SYSRSn

16.7.4.8 CPU2TOCPU3INTIPCSET_j Register (Offset = 2000h + formula) [Reset = 00000000h]

CPU2TOCPU3INTIPCSET_j is shown in Figure 16-28 and described in Table 16-36.

Return to the Summary Table.

Set CPU2TOCPU3INTIPCSET Register

Offset = 2000h + (j * 800h); where j = 0h to 3h

Figure 16-28. CPU2TOCPU3INTIPCSET_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-36. CPU2TOCPU3INTIPCSET_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit sets the IPC31 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit sets the IPC30 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit sets the IPC29 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit sets the IPC28 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit sets the IPC27 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit sets the IPC26 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit sets the IPC25 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
24	IPC24	R-0/W1S	0h	Writing 1 to this bit sets the IPC24 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit sets the IPC23 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn

Table 16-36. CPU2TOCPU3INTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R-0/W1S	0h	Writing 1 to this bit sets the IPC22 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit sets the IPC21 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit sets the IPC20 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit sets the IPC19 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit sets the IPC18 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit sets the IPC17 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit sets the IPC16 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit sets the IPC15 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit sets the IPC14 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit sets the IPC13 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit sets the IPC12 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit sets the IPC11 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
10	IPC10	R-0/W1S	0h	Writing 1 to this bit sets the IPC10 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit sets the IPC9 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit sets the IPC8 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit sets the IPC7 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit sets the IPC6 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn

Table 16-36. CPU2TOCPU3INTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R-0/W1S	0h	Writing 1 to this bit sets the IPC5 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit sets the IPC4 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit sets the IPC3 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit sets the IPC2 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit sets the IPC1 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit sets the IPC0 event flag for the remote CPU. Writing 0 has no effect. Notes: [1] IPC event flags 0 will trigger interrupts in the receiving CPU via the PIPE. Reset type: CPU2.SYSRSn

16.7.4.9 CPU2TOCPU3INTIPCCLR_j Register (Offset = 2004h + formula) [Reset = 00000000h]

CPU2TOCPU3INTIPCCLR_j is shown in Figure 16-29 and described in Table 16-37.

Return to the Summary Table.

Clear CPU2TOCPU3IPCFLG register

Offset = 2004h + (j * 800h); where j = 0h to 3h

Figure 16-29. CPU2TOCPU3INTIPCCLR_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-37. CPU2TOCPU3INTIPCCLR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC31 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC30 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC29 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC28 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC27 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC26 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC25 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn

Table 16-37. CPU2TOCPU3INTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	IPC24	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC24 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC23 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
22	IPC22	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC22 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC21 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC20 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC19 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC18 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC17 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC16 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC15 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC14 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC13 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC12 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC11 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn

Table 16-37. CPU2TOCPU3INTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	IPC10	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC10 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC9 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC8 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC7 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC6 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
5	IPC5	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC5 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC4 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC3 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC2 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC1 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOCPU3IPCFLG.IPC0 event flag for CPU3. Writing 0 has no effect. Reset type: CPU2.SYSRSn

16.7.4.10 CPU2TOCPU3INTIPCFLG_j Register (Offset = 2008h + formula) [Reset = 00000000h]

CPU2TOCPU3INTIPCFLG_j is shown in Figure 16-30 and described in Table 16-38.

Return to the Summary Table.

CPU2TOCPU3INTIPCFLG Register

Offset = 2008h + (j * 800h); where j = 0h to 3h

Figure 16-30. CPU2TOCPU3INTIPCFLG_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-38. CPU2TOCPU3INTIPCFLG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R	0h	0: No IPC31 event request to CPU3 1: IPC31 event request to CPU3 Reset type: CPUx.SYSRSn
30	IPC30	R	0h	0: No IPC30 event request to CPU3 1: IPC30 event request to CPU3 Reset type: CPUx.SYSRSn
29	IPC29	R	0h	0: No IPC29 event request to CPU3 1: IPC29 event request to CPU3 Reset type: CPUx.SYSRSn
28	IPC28	R	0h	0: No IPC28 event request to CPU3 1: IPC28 event request to CPU3 Reset type: CPUx.SYSRSn
27	IPC27	R	0h	0: No IPC27 event request to CPU3 1: IPC27 event request to CPU3 Reset type: CPUx.SYSRSn
26	IPC26	R	0h	0: No IPC26 event request to CPU3 1: IPC26 event request to CPU3 Reset type: CPUx.SYSRSn
25	IPC25	R	0h	0: No IPC25 event request to CPU3 1: IPC25 event request to CPU3 Reset type: CPUx.SYSRSn
24	IPC24	R	0h	0: No IPC24 event request to CPU3 1: IPC24 event request to CPU3 Reset type: CPUx.SYSRSn
23	IPC23	R	0h	0: No IPC23 event request to CPU3 1: IPC23 event request to CPU3 Reset type: CPUx.SYSRSn

Table 16-38. CPU2TOCPU3INTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R	0h	0: No IPC22 event request to CPU3 1: IPC22 event request to CPU3 Reset type: CPUx.SYSRSn
21	IPC21	R	0h	0: No IPC21 event request to CPU3 1: IPC21 event request to CPU3 Reset type: CPUx.SYSRSn
20	IPC20	R	0h	0: No IPC20 event request to CPU3 1: IPC20 event request to CPU3 Reset type: CPUx.SYSRSn
19	IPC19	R	0h	0: No IPC19 event request to CPU3 1: IPC19 event request to CPU3 Reset type: CPUx.SYSRSn
18	IPC18	R	0h	0: No IPC18 event request to CPU3 1: IPC18 event request to CPU3 Reset type: CPUx.SYSRSn
17	IPC17	R	0h	0: No IPC17 event request to CPU3 1: IPC17 event request to CPU3 Reset type: CPUx.SYSRSn
16	IPC16	R	0h	0: No IPC16 event request to CPU3 1: IPC16 event request to CPU3 Reset type: CPUx.SYSRSn
15	IPC15	R	0h	0: No IPC15 event request to CPU3 1: IPC15 event request to CPU3 Reset type: CPUx.SYSRSn
14	IPC14	R	0h	0: No IPC14 event request to CPU3 1: IPC14 event request to CPU3 Reset type: CPUx.SYSRSn
13	IPC13	R	0h	0: No IPC13 event request to CPU3 1: IPC13 event request to CPU3 Reset type: CPUx.SYSRSn
12	IPC12	R	0h	0: No IPC12 event request to CPU3 1: IPC12 event request to CPU3 Reset type: CPUx.SYSRSn
11	IPC11	R	0h	0: No IPC11 event request to CPU3 1: IPC11 event request to CPU3 Reset type: CPUx.SYSRSn
10	IPC10	R	0h	0: No IPC10 event request to CPU3 1: IPC10 event request to CPU3 Reset type: CPUx.SYSRSn
9	IPC9	R	0h	0: No IPC9 event request to CPU3 1: IPC9 event request to CPU3 Reset type: CPUx.SYSRSn
8	IPC8	R	0h	0: No IPC8 event request to CPU3 1: IPC8 event request to CPU3 Reset type: CPUx.SYSRSn
7	IPC7	R	0h	0: No IPC7 event request to CPU3 1: IPC7 event request to CPU3 Reset type: CPUx.SYSRSn
6	IPC6	R	0h	0: No IPC6 event request to CPU3 1: IPC6 event request to CPU3 Reset type: CPUx.SYSRSn

Table 16-38. CPU2TOCPU3INTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R	0h	0: No IPC5 event request to CPU3 1: IPC5 event request to CPU3 Reset type: CPUx.SYSRSn
4	IPC4	R	0h	0: No IPC4 event request to CPU3 1: IPC4 event request to CPU3 Reset type: CPUx.SYSRSn
3	IPC3	R	0h	0: No IPC3 event request to CPU3 1: IPC3 event request to CPU3 Reset type: CPUx.SYSRSn
2	IPC2	R	0h	0: No IPC2 event request to CPU3 1: IPC2 event request to CPU3 Reset type: CPUx.SYSRSn
1	IPC1	R	0h	0: No IPC1 event request to CPU3 1: IPC1 event request to CPU3 Reset type: CPUx.SYSRSn
0	IPC0	R	0h	0: No IPC0 event request to CPU3 1: IPC0 event request to CPU3 Reset type: CPUx.SYSRSn

16.7.4.11 CPU2TOCPU3INTIPCSENDCOM_j Register (Offset = 2010h + formula) [Reset = 00000000h]

CPU2TOCPU3INTIPCSENDCOM_j is shown in Figure 16-31 and described in Table 16-39.

Return to the Summary Table.

CPU2 to CPU3 IPC Command

Offset = 2010h + (j * 800h); where j = 0h to 3h

Figure 16-31. CPU2TOCPU3INTIPCSENDCOM_j Register

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

Table 16-39. CPU2TOCPU3INTIPCSENDCOM_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMMAND	R/W	0h	This is a general purpose register used to send software-defined commands to from CPU2 to CPU3 Reset type: CPU2.SYSRSn

16.7.4.12 CPU2TOCPU3INTIPCSENDADDR_j Register (Offset = 2014h + formula) [Reset = 00000000h]

CPU2TOCPU3INTIPCSENDADDR_j is shown in Figure 16-32 and described in Table 16-40.

Return to the Summary Table.

CPU2 to CPU3 IPC Address

Offset = 2014h + (j * 800h); where j = 0h to 3h

Figure 16-32. CPU2TOCPU3INTIPCSENDADDR_j Register

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

Table 16-40. CPU2TOCPU3INTIPCSENDADDR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R/W	0h	This is a general purpose register used to send software-defined Address to from CPU2 to CPU3 Reset type: CPU2.SYSRSn

16.7.4.13 CPU2TOCPU3INTIPCSENDDATA_j Register (Offset = 2018h + formula) [Reset = 00000000h]

CPU2TOCPU3INTIPCSENDDATA_j is shown in Figure 16-33 and described in Table 16-41.

Return to the Summary Table.

CPU2 to CPU3 IPC Data

Offset = 2018h + (j * 800h); where j = 0h to 3h

Figure 16-33. CPU2TOCPU3INTIPCSENDDATA_j Register

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

Table 16-41. CPU2TOCPU3INTIPCSENDDATA_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/W	0h	This is a general purpose register used to send software-defined Data to from CPU2 to CPU3 Reset type: CPU2.SYSRSn

16.7.4.14 CPU3TOCPU2INTREMOTEREPLY_j Register (Offset = 201Ch + formula) [Reset = 00000000h]

CPU3TOCPU2INTREMOTEREPLY_j is shown in Figure 16-34 and described in Table 16-42.

Return to the Summary Table.

Reply from CPU3 to CPU2TOCPU3IPCSENDCOM command request

Offset = 201Ch + (j * 800h); where j = 0h to 3h

Figure 16-34. CPU3TOCPU2INTREMOTEREPLY_j Register

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

Table 16-42. CPU3TOCPU2INTREMOTEREPLY_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REPLY	R	0h	Refelects the state of CPU3TOCPU2INT IPCREPLY register Reset type: CPU2.SYSRSn

16.7.4.15 CPU2TOHSMINTIPCSET_j Register (Offset = 6000h + formula) [Reset = 00000000h]

CPU2TOHSMINTIPCSET_j is shown in Figure 16-35 and described in Table 16-43.

Return to the Summary Table.

Set CPU2TOHSMINTIPCSET Register

Offset = 6000h + (j * 1000h); where j = 0h to 1h

Figure 16-35. CPU2TOHSMINTIPCSET_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-43. CPU2TOHSMINTIPCSET_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit sets the IPC31 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit sets the IPC30 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit sets the IPC29 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit sets the IPC28 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit sets the IPC27 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit sets the IPC26 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit sets the IPC25 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
24	IPC24	R-0/W1S	0h	Writing 1 to this bit sets the IPC24 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit sets the IPC23 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn

Table 16-43. CPU2TOHSMINTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R-0/W1S	0h	Writing 1 to this bit sets the IPC22 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit sets the IPC21 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit sets the IPC20 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit sets the IPC19 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit sets the IPC18 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit sets the IPC17 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit sets the IPC16 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit sets the IPC15 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit sets the IPC14 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit sets the IPC13 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit sets the IPC12 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit sets the IPC11 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
10	IPC10	R-0/W1S	0h	Writing 1 to this bit sets the IPC10 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit sets the IPC9 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit sets the IPC8 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit sets the IPC7 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit sets the IPC6 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn

Table 16-43. CPU2TOHSMINTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R-0/W1S	0h	Writing 1 to this bit sets the IPC5 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit sets the IPC4 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit sets the IPC3 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit sets the IPC2 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit sets the IPC1 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPU2.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit sets the IPC0 event flag for the remote CPU. Writing 0 has no effect. Notes: [1] IPC event flags 0 will trigger interrupts in the receiving CPU via the PIPE. Reset type: CPU2.SYSRSn

16.7.4.16 CPU2TOHSMINTIPCCLR_j Register (Offset = 6004h + formula) [Reset = 00000000h]

CPU2TOHSMINTIPCCLR_j is shown in Figure 16-36 and described in Table 16-44.

Return to the Summary Table.

Clear CPU2TOHSMIPCFLG register

Offset = 6004h + (j * 1000h); where j = 0h to 1h

Figure 16-36. CPU2TOHSMINTIPCCLR_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-44. CPU2TOHSMINTIPCCLR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC31 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC30 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC29 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC28 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC27 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC26 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC25 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn

Table 16-44. CPU2TOHSMINTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	IPC24	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC24 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC23 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
22	IPC22	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC22 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC21 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC20 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC19 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC18 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC17 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC16 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC15 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC14 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC13 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC12 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC11 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn

Table 16-44. CPU2TOHSMINTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	IPC10	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC10 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC9 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC8 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC7 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC6 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
5	IPC5	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC5 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC4 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC3 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC2 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC1 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit clear the CPU2TOHSMIPCFLG.IPC0 event flag for HSM. Writing 0 has no effect. Reset type: CPU2.SYSRSn

16.7.4.17 CPU2TOHSMINTIPCFLG_j Register (Offset = 6008h + formula) [Reset = 00000000h]

CPU2TOHSMINTIPCFLG_j is shown in Figure 16-37 and described in Table 16-45.

Return to the Summary Table.

CPU2TOHSMINTIPCFLG Register

Offset = 6008h + (j * 1000h); where j = 0h to 1h

Figure 16-37. CPU2TOHSMINTIPCFLG_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-45. CPU2TOHSMINTIPCFLG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R	0h	0: No IPC31 event request to CPU2 1: IPC31 event request to CPU2 Reset type: CPU2.SYSRSn
30	IPC30	R	0h	0: No IPC30 event request to CPU2 1: IPC30 event request to CPU2 Reset type: CPU2.SYSRSn
29	IPC29	R	0h	0: No IPC29 event request to CPU2 1: IPC29 event request to CPU2 Reset type: CPU2.SYSRSn
28	IPC28	R	0h	0: No IPC28 event request to CPU2 1: IPC28 event request to CPU2 Reset type: CPU2.SYSRSn
27	IPC27	R	0h	0: No IPC27 event request to CPU2 1: IPC27 event request to CPU2 Reset type: CPU2.SYSRSn
26	IPC26	R	0h	0: No IPC26 event request to CPU2 1: IPC26 event request to CPU2 Reset type: CPU2.SYSRSn
25	IPC25	R	0h	0: No IPC25 event request to CPU2 1: IPC25 event request to CPU2 Reset type: CPU2.SYSRSn
24	IPC24	R	0h	0: No IPC24 event request to CPU2 1: IPC24 event request to CPU2 Reset type: CPU2.SYSRSn
23	IPC23	R	0h	0: No IPC23 event request to CPU2 1: IPC23 event request to CPU2 Reset type: CPU2.SYSRSn

Table 16-45. CPU2TOHSMINTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R	0h	0: No IPC22 event request to CPU2 1: IPC22 event request to CPU2 Reset type: CPU2.SYSRSn
21	IPC21	R	0h	0: No IPC21 event request to CPU2 1: IPC21 event request to CPU2 Reset type: CPU2.SYSRSn
20	IPC20	R	0h	0: No IPC20 event request to CPU2 1: IPC20 event request to CPU2 Reset type: CPU2.SYSRSn
19	IPC19	R	0h	0: No IPC19 event request to CPU2 1: IPC19 event request to CPU2 Reset type: CPU2.SYSRSn
18	IPC18	R	0h	0: No IPC18 event request to CPU2 1: IPC18 event request to CPU2 Reset type: CPU2.SYSRSn
17	IPC17	R	0h	0: No IPC17 event request to CPU2 1: IPC17 event request to CPU2 Reset type: CPU2.SYSRSn
16	IPC16	R	0h	0: No IPC16 event request to CPU2 1: IPC16 event request to CPU2 Reset type: CPU2.SYSRSn
15	IPC15	R	0h	0: No IPC15 event request to CPU2 1: IPC15 event request to CPU2 Reset type: CPU2.SYSRSn
14	IPC14	R	0h	0: No IPC14 event request to CPU2 1: IPC14 event request to CPU2 Reset type: CPU2.SYSRSn
13	IPC13	R	0h	0: No IPC13 event request to CPU2 1: IPC13 event request to CPU2 Reset type: CPU2.SYSRSn
12	IPC12	R	0h	0: No IPC12 event request to CPU2 1: IPC12 event request to CPU2 Reset type: CPU2.SYSRSn
11	IPC11	R	0h	0: No IPC11 event request to CPU2 1: IPC11 event request to CPU2 Reset type: CPU2.SYSRSn
10	IPC10	R	0h	0: No IPC10 event request to CPU2 1: IPC10 event request to CPU2 Reset type: CPU2.SYSRSn
9	IPC9	R	0h	0: No IPC9 event request to CPU2 1: IPC9 event request to CPU2 Reset type: CPU2.SYSRSn
8	IPC8	R	0h	0: No IPC8 event request to CPU2 1: IPC8 event request to CPU2 Reset type: CPU2.SYSRSn
7	IPC7	R	0h	0: No IPC7 event request to CPU2 1: IPC7 event request to CPU2 Reset type: CPU2.SYSRSn
6	IPC6	R	0h	0: No IPC6 event request to CPU2 1: IPC6 event request to CPU2 Reset type: CPU2.SYSRSn

Table 16-45. CPU2TOHSMINTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R	0h	0: No IPC5 event request to CPU2 1: IPC5 event request to CPU2 Reset type: CPU2.SYSRSn
4	IPC4	R	0h	0: No IPC4 event request to CPU2 1: IPC4 event request to CPU2 Reset type: CPU2.SYSRSn
3	IPC3	R	0h	0: No IPC3 event request to CPU2 1: IPC3 event request to CPU2 Reset type: CPU2.SYSRSn
2	IPC2	R	0h	0: No IPC2 event request to CPU2 1: IPC2 event request to CPU2 Reset type: CPU2.SYSRSn
1	IPC1	R	0h	0: No IPC1 event request to CPU2 1: IPC1 event request to CPU2 Reset type: CPU2.SYSRSn
0	IPC0	R	0h	0: No IPC0 event request to CPU2 1: IPC0 event request to CPU2 Reset type: CPU2.SYSRSn

16.7.5 CPU3_IPC_SEND_REGS Registers

Table 16-46 lists the memory-mapped registers for the CPU3_IPC_SEND_REGS registers. All register offset addresses not listed in Table 16-46 should be considered as reserved locations and the register contents should not be modified.

Table 16-46. CPU3_IPC_SEND_REGS Registers

Offset	Acronym	Register Name	Protection
0h + formula	CPU3TOCPU1INTIPCSET_j	CPU3TOCPU1INTIPCSET Register	
4h + formula	CPU3TOCPU1INTIPCCLR_j	CPU3TOCPU1INTIPCCLR Register	
8h + formula	CPU3TOCPU1INTIPCFLG_j	CPU3TOCPU1INTIPCFLG Register	
10h + formula	CPU3TOCPU1INTIPCSENDCOM_j	CPU3TOCPU1INTIPCSENDCOM Register	
14h + formula	CPU3TOCPU1INTIPCSENDADDR_j	CPU3TOCPU1INTIPCSENDADDR Register	
18h + formula	CPU3TOCPU1INTIPCSENDDATA_j	CPU3TOCPU1INTIPCSENDDATA Register	
1Ch + formula	CPU1TOCPU3INTREMOTEREPLY_j	CPU1TOCPU3INTREMOTEREPLY Register	
2000h + formula	CPU3TOCPU2INTIPCSET_j	CPU3TOCPU2INTIPCSET Register	
2004h + formula	CPU3TOCPU2INTIPCCLR_j	CPU3TOCPU2INTIPCCLR Register	
2008h + formula	CPU3TOCPU2INTIPCFLG_j	CPU3TOCPU2INTIPCFLG Register	
2010h + formula	CPU3TOCPU2INTIPCSENDCOM_j	CPU3TOCPU2INTIPCSENDCOM Register	
2014h + formula	CPU3TOCPU2INTIPCSENDADDR_j	CPU3TOCPU2INTIPCSENDADDR Register	
2018h + formula	CPU3TOCPU2INTIPCSENDDATA_j	CPU3TOCPU2INTIPCSENDDATA Register	
201Ch + formula	CPU2TOCPU3INTREMOTEREPLY_j	CPU2TOCPU3INTREMOTEREPLY Register	
6000h + formula	CPU3TOHSMINTIPCSET_j	CPU3TOHSMINTIPCSET Register	
6004h + formula	CPU3TOHSMINTIPCCLR_j	CPU3TOHSMINTIPCCLR Register	
6008h + formula	CPU3TOHSMINTIPCFLG_j	CPU3TOHSMINTIPCFLG Register	

Complex bit access types are encoded to fit into small table cells. Table 16-47 shows the codes that are used for access types in this section.

Table 16-47. CPU3_IPC_SEND_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		

Table 16-47. CPU3_IPC_SEND_REGS Access Type Codes (continued)

Access Type	Code	Description
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

16.7.5.1 CPU3TOCPU1INTIPCSET_j Register (Offset = 0h + formula) [Reset = 00000000h]

CPU3TOCPU1INTIPCSET_j is shown in Figure 16-38 and described in Table 16-48.

Return to the Summary Table.

Set CPU1TOCPU3IPCFLG register

Offset = 0h + (j * 800h); where j = 0h to 3h

Figure 16-38. CPU3TOCPU1INTIPCSET_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-48. CPU3TOCPU1INTIPCSET_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit sets the IPC31 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit sets the IPC30 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit sets the IPC29 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit sets the IPC28 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit sets the IPC27 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit sets the IPC26 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit sets the IPC25 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
24	IPC24	R-0/W1S	0h	Writing 1 to this bit sets the IPC24 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit sets the IPC23 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn

Table 16-48. CPU3TOCPU1INTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R-0/W1S	0h	Writing 1 to this bit sets the IPC22 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit sets the IPC21 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit sets the IPC20 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit sets the IPC19 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit sets the IPC18 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit sets the IPC17 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit sets the IPC16 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit sets the IPC15 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit sets the IPC14 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit sets the IPC13 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit sets the IPC12 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit sets the IPC11 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
10	IPC10	R-0/W1S	0h	Writing 1 to this bit sets the IPC10 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit sets the IPC9 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit sets the IPC8 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit sets the IPC7 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit sets the IPC6 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn

Table 16-48. CPU3TOCPU1INTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R-0/W1S	0h	Writing 1 to this bit sets the IPC5 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit sets the IPC4 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit sets the IPC3 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit sets the IPC2 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit sets the IPC1 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit sets the IPC0 event flag for the remote CPU. Writing 0 has no effect. Notes: [1] IPC event flags 0 will trigger interrupts in the receiving CPU via the PIPE. Reset type: CPUx.SYSRSn

16.7.5.2 CPU3TOCPU1INTIPCCLR_j Register (Offset = 4h + formula) [Reset = 00000000h]

CPU3TOCPU1INTIPCCLR_j is shown in Figure 16-39 and described in Table 16-49.

Return to the Summary Table.

Clear CPU3TOCPU1IPCFLG register

Offset = 4h + (j * 800h); where j = 0h to 3h

Figure 16-39. CPU3TOCPU1INTIPCCLR_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-49. CPU3TOCPU1INTIPCCLR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC31 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC30 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC29 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC28 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC27 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC26 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC25 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn

Table 16-49. CPU3TOCPU1INTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	IPC24	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC24 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC23 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
22	IPC22	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC22 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC21 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC20 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC19 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC18 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC17 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC16 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC15 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC14 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC13 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC12 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC11 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn

Table 16-49. CPU3TOCPU1INTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	IPC10	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC10 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC9 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC8 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC7 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC6 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
5	IPC5	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC5 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC4 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC3 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC2 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC1 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU1IPCFLG.IPC0 event flag for CPU1. Writing 0 has no effect. Reset type: CPUx.SYSRSn

16.7.5.3 CPU3TOCPU1INTIPCFLG_j Register (Offset = 8h + formula) [Reset = 00000000h]

CPU3TOCPU1INTIPCFLG_j is shown in Figure 16-40 and described in Table 16-50.

Return to the Summary Table.

CPU3TOCPU1INTIPCFLG Register

Offset = 8h + (j * 800h); where j = 0h to 3h

Figure 16-40. CPU3TOCPU1INTIPCFLG_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-50. CPU3TOCPU1INTIPCFLG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R	0h	0: No IPC31 event request to CPU1 1: IPC31 event request to CPU1 Reset type: CPU1.SYSRSn
30	IPC30	R	0h	0: No IPC30 event request to CPU1 1: IPC30 event request to CPU1 Reset type: CPU1.SYSRSn
29	IPC29	R	0h	0: No IPC29 event request to CPU1 1: IPC29 event request to CPU1 Reset type: CPU1.SYSRSn
28	IPC28	R	0h	0: No IPC28 event request to CPU1 1: IPC28 event request to CPU1 Reset type: CPU1.SYSRSn
27	IPC27	R	0h	0: No IPC27 event request to CPU1 1: IPC27 event request to CPU1 Reset type: CPU1.SYSRSn
26	IPC26	R	0h	0: No IPC26 event request to CPU1 1: IPC26 event request to CPU1 Reset type: CPU1.SYSRSn
25	IPC25	R	0h	0: No IPC25 event request to CPU1 1: IPC25 event request to CPU1 Reset type: CPU1.SYSRSn
24	IPC24	R	0h	0: No IPC24 event request to CPU1 1: IPC24 event request to CPU1 Reset type: CPU1.SYSRSn
23	IPC23	R	0h	0: No IPC23 event request to CPU1 1: IPC23 event request to CPU1 Reset type: CPU1.SYSRSn

Table 16-50. CPU3TOCPU1INTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R	0h	0: No IPC22 event request to CPU1 1: IPC22 event request to CPU1 Reset type: CPU1.SYSRSn
21	IPC21	R	0h	0: No IPC21 event request to CPU1 1: IPC21 event request to CPU1 Reset type: CPU1.SYSRSn
20	IPC20	R	0h	0: No IPC20 event request to CPU1 1: IPC20 event request to CPU1 Reset type: CPU1.SYSRSn
19	IPC19	R	0h	0: No IPC19 event request to CPU1 1: IPC19 event request to CPU1 Reset type: CPU1.SYSRSn
18	IPC18	R	0h	0: No IPC18 event request to CPU1 1: IPC18 event request to CPU1 Reset type: CPU1.SYSRSn
17	IPC17	R	0h	0: No IPC17 event request to CPU1 1: IPC17 event request to CPU1 Reset type: CPU1.SYSRSn
16	IPC16	R	0h	0: No IPC16 event request to CPU1 1: IPC16 event request to CPU1 Reset type: CPU1.SYSRSn
15	IPC15	R	0h	0: No IPC15 event request to CPU1 1: IPC15 event request to CPU1 Reset type: CPU1.SYSRSn
14	IPC14	R	0h	0: No IPC14 event request to CPU1 1: IPC14 event request to CPU1 Reset type: CPU1.SYSRSn
13	IPC13	R	0h	0: No IPC13 event request to CPU1 1: IPC13 event request to CPU1 Reset type: CPU1.SYSRSn
12	IPC12	R	0h	0: No IPC12 event request to CPU1 1: IPC12 event request to CPU1 Reset type: CPU1.SYSRSn
11	IPC11	R	0h	0: No IPC11 event request to CPU1 1: IPC11 event request to CPU1 Reset type: CPU1.SYSRSn
10	IPC10	R	0h	0: No IPC10 event request to CPU1 1: IPC10 event request to CPU1 Reset type: CPU1.SYSRSn
9	IPC9	R	0h	0: No IPC9 event request to CPU1 1: IPC9 event request to CPU1 Reset type: CPU1.SYSRSn
8	IPC8	R	0h	0: No IPC8 event request to CPU1 1: IPC8 event request to CPU1 Reset type: CPU1.SYSRSn
7	IPC7	R	0h	0: No IPC7 event request to CPU1 1: IPC7 event request to CPU1 Reset type: CPU1.SYSRSn
6	IPC6	R	0h	0: No IPC6 event request to CPU1 1: IPC6 event request to CPU1 Reset type: CPU1.SYSRSn

Table 16-50. CPU3TOCPU1INTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R	0h	0: No IPC5 event request to CPU1 1: IPC5 event request to CPU1 Reset type: CPU1.SYSRSn
4	IPC4	R	0h	0: No IPC4 event request to CPU1 1: IPC4 event request to CPU1 Reset type: CPU1.SYSRSn
3	IPC3	R	0h	0: No IPC3 event request to CPU1 1: IPC3 event request to CPU1 Reset type: CPU1.SYSRSn
2	IPC2	R	0h	0: No IPC2 event request to CPU1 1: IPC2 event request to CPU1 Reset type: CPU1.SYSRSn
1	IPC1	R	0h	0: No IPC1 event request to CPU1 1: IPC1 event request to CPU1 Reset type: CPU1.SYSRSn
0	IPC0	R	0h	0: No IPC0 event request to CPU1 1: IPC0 event request to CPU1 Reset type: CPU1.SYSRSn

16.7.5.4 CPU3TOCPU1INTIPCSENDCOM_j Register (Offset = 10h + formula) [Reset = 00000000h]

CPU3TOCPU1INTIPCSENDCOM_j is shown in Figure 16-41 and described in Table 16-51.

Return to the Summary Table.

CPU3 to CPU1 IPC Command

Offset = 10h + (j * 800h); where j = 0h to 3h

Figure 16-41. CPU3TOCPU1INTIPCSENDCOM_j Register

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

Table 16-51. CPU3TOCPU1INTIPCSENDCOM_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMMAND	R/W	0h	This is a general purpose register used to send software-defined commands to from CPU3 to CPU1 Reset type: CPUx.SYSRSn

16.7.5.5 CPU3TOCPU1INTIPCSENDADDR_j Register (Offset = 14h + formula) [Reset = 00000000h]

CPU3TOCPU1INTIPCSENDADDR_j is shown in Figure 16-42 and described in Table 16-52.

Return to the Summary Table.

CPU3 to CPU1 IPC Address

Offset = 14h + (j * 800h); where j = 0h to 3h

Figure 16-42. CPU3TOCPU1INTIPCSENDADDR_j Register

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

Table 16-52. CPU3TOCPU1INTIPCSENDADDR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R/W	0h	This is a general purpose register used to send software-defined Address to from CPU3 to CPU1 Reset type: CPUx.SYSRSn

16.7.5.6 CPU3TOCPU1INTIPCSENDDATA_j Register (Offset = 18h + formula) [Reset = 00000000h]

CPU3TOCPU1INTIPCSENDDATA_j is shown in Figure 16-43 and described in Table 16-53.

Return to the Summary Table.

CPU3 to CPU1 IPC Data

Offset = 18h + (j * 800h); where j = 0h to 3h

Figure 16-43. CPU3TOCPU1INTIPCSENDDATA_j Register

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

Table 16-53. CPU3TOCPU1INTIPCSENDDATA_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/W	0h	This is a general purpose register used to send software-defined Data to from CPU3 to CPU1 Reset type: CPUx.SYSRSn

16.7.5.7 CPU1TOCPU3INTREMOTEREPLY_j Register (Offset = 1Ch + formula) [Reset = 00000000h]

CPU1TOCPU3INTREMOTEREPLY_j is shown in Figure 16-44 and described in Table 16-54.

Return to the Summary Table.

Reply from CPU1 to CPU3TOCPU1IPCSENDCOM command request

Offset = 1Ch + (j * 800h); where j = 0h to 3h

Figure 16-44. CPU1TOCPU3INTREMOTEREPLY_j Register

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

Table 16-54. CPU1TOCPU3INTREMOTEREPLY_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REPLY	R	0h	Refelects the state of CPU1TOCPU3INT IPCREPLY register Reset type: CPUx.SYSRSn

16.7.5.8 CPU3TOCPU2INTIPCSET_j Register (Offset = 2000h + formula) [Reset = 00000000h]

CPU3TOCPU2INTIPCSET_j is shown in Figure 16-45 and described in Table 16-55.

Return to the Summary Table.

Set CPU3TOCPU2INTIPCSET Register

Offset = 2000h + (j * 800h); where j = 0h to 3h

Figure 16-45. CPU3TOCPU2INTIPCSET_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-55. CPU3TOCPU2INTIPCSET_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit sets the IPC31 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit sets the IPC30 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit sets the IPC29 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit sets the IPC28 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit sets the IPC27 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit sets the IPC26 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit sets the IPC25 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
24	IPC24	R-0/W1S	0h	Writing 1 to this bit sets the IPC24 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit sets the IPC23 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn

Table 16-55. CPU3TOCPU2INTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R-0/W1S	0h	Writing 1 to this bit sets the IPC22 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit sets the IPC21 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit sets the IPC20 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit sets the IPC19 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit sets the IPC18 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit sets the IPC17 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit sets the IPC16 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit sets the IPC15 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit sets the IPC14 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit sets the IPC13 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit sets the IPC12 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit sets the IPC11 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
10	IPC10	R-0/W1S	0h	Writing 1 to this bit sets the IPC10 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit sets the IPC9 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit sets the IPC8 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit sets the IPC7 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit sets the IPC6 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn

Table 16-55. CPU3TOCPU2INTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R-0/W1S	0h	Writing 1 to this bit sets the IPC5 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit sets the IPC4 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit sets the IPC3 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit sets the IPC2 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit sets the IPC1 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit sets the IPC0 event flag for the remote CPU. Writing 0 has no effect. Notes: [1] IPC event flags 0 will trigger interrupts in the receiving CPU via the PIPE. Reset type: CPUx.SYSRSn

16.7.5.9 CPU3TOCPU2INTIPCCLR_j Register (Offset = 2004h + formula) [Reset = 00000000h]

CPU3TOCPU2INTIPCCLR_j is shown in Figure 16-46 and described in Table 16-56.

Return to the Summary Table.

Clear CPU3TOCPU2IPCFLG register

Offset = 2004h + (j * 800h); where j = 0h to 3h

Figure 16-46. CPU3TOCPU2INTIPCCLR_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-56. CPU3TOCPU2INTIPCCLR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC31 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC30 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC29 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC28 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC27 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC26 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC25 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn

Table 16-56. CPU3TOCPU2INTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	IPC24	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC24 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC23 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
22	IPC22	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC22 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC21 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC20 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC19 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC18 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC17 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC16 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC15 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC14 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC13 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC12 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC11 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn

Table 16-56. CPU3TOCPU2INTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	IPC10	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC10 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC9 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC8 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC7 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC6 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
5	IPC5	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC5 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC4 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC3 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC2 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC1 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOCPU2IPCFLG.IPC0 event flag for CPU2. Writing 0 has no effect. Reset type: CPUx.SYSRSn

16.7.5.10 CPU3TOCPU2INTIPCFLG_j Register (Offset = 2008h + formula) [Reset = 00000000h]

CPU3TOCPU2INTIPCFLG_j is shown in Figure 16-47 and described in Table 16-57.

Return to the Summary Table.

CPU3TOCPU2INTIPCFLG Register

Offset = 2008h + (j * 800h); where j = 0h to 3h

Figure 16-47. CPU3TOCPU2INTIPCFLG_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-57. CPU3TOCPU2INTIPCFLG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R	0h	0: No IPC31 event request to CPU2 1: IPC31 event request to CPU2 Reset type: CPU2.SYSRSn
30	IPC30	R	0h	0: No IPC30 event request to CPU2 1: IPC30 event request to CPU2 Reset type: CPU2.SYSRSn
29	IPC29	R	0h	0: No IPC29 event request to CPU2 1: IPC29 event request to CPU2 Reset type: CPU2.SYSRSn
28	IPC28	R	0h	0: No IPC28 event request to CPU2 1: IPC28 event request to CPU2 Reset type: CPU2.SYSRSn
27	IPC27	R	0h	0: No IPC27 event request to CPU2 1: IPC27 event request to CPU2 Reset type: CPU2.SYSRSn
26	IPC26	R	0h	0: No IPC26 event request to CPU2 1: IPC26 event request to CPU2 Reset type: CPU2.SYSRSn
25	IPC25	R	0h	0: No IPC25 event request to CPU2 1: IPC25 event request to CPU2 Reset type: CPU2.SYSRSn
24	IPC24	R	0h	0: No IPC24 event request to CPU2 1: IPC24 event request to CPU2 Reset type: CPU2.SYSRSn
23	IPC23	R	0h	0: No IPC23 event request to CPU2 1: IPC23 event request to CPU2 Reset type: CPU2.SYSRSn

Table 16-57. CPU3TOCPU2INTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R	0h	0: No IPC22 event request to CPU2 1: IPC22 event request to CPU2 Reset type: CPU2.SYSRSn
21	IPC21	R	0h	0: No IPC21 event request to CPU2 1: IPC21 event request to CPU2 Reset type: CPU2.SYSRSn
20	IPC20	R	0h	0: No IPC20 event request to CPU2 1: IPC20 event request to CPU2 Reset type: CPU2.SYSRSn
19	IPC19	R	0h	0: No IPC19 event request to CPU2 1: IPC19 event request to CPU2 Reset type: CPU2.SYSRSn
18	IPC18	R	0h	0: No IPC18 event request to CPU2 1: IPC18 event request to CPU2 Reset type: CPU2.SYSRSn
17	IPC17	R	0h	0: No IPC17 event request to CPU2 1: IPC17 event request to CPU2 Reset type: CPU2.SYSRSn
16	IPC16	R	0h	0: No IPC16 event request to CPU2 1: IPC16 event request to CPU2 Reset type: CPU2.SYSRSn
15	IPC15	R	0h	0: No IPC15 event request to CPU2 1: IPC15 event request to CPU2 Reset type: CPU2.SYSRSn
14	IPC14	R	0h	0: No IPC14 event request to CPU2 1: IPC14 event request to CPU2 Reset type: CPU2.SYSRSn
13	IPC13	R	0h	0: No IPC13 event request to CPU2 1: IPC13 event request to CPU2 Reset type: CPU2.SYSRSn
12	IPC12	R	0h	0: No IPC12 event request to CPU2 1: IPC12 event request to CPU2 Reset type: CPU2.SYSRSn
11	IPC11	R	0h	0: No IPC11 event request to CPU2 1: IPC11 event request to CPU2 Reset type: CPU2.SYSRSn
10	IPC10	R	0h	0: No IPC10 event request to CPU2 1: IPC10 event request to CPU2 Reset type: CPU2.SYSRSn
9	IPC9	R	0h	0: No IPC9 event request to CPU2 1: IPC9 event request to CPU2 Reset type: CPU2.SYSRSn
8	IPC8	R	0h	0: No IPC8 event request to CPU2 1: IPC8 event request to CPU2 Reset type: CPU2.SYSRSn
7	IPC7	R	0h	0: No IPC7 event request to CPU2 1: IPC7 event request to CPU2 Reset type: CPU2.SYSRSn
6	IPC6	R	0h	0: No IPC6 event request to CPU2 1: IPC6 event request to CPU2 Reset type: CPU2.SYSRSn

Table 16-57. CPU3TOCPU2INTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R	0h	0: No IPC5 event request to CPU2 1: IPC5 event request to CPU2 Reset type: CPU2.SYSRSn
4	IPC4	R	0h	0: No IPC4 event request to CPU2 1: IPC4 event request to CPU2 Reset type: CPU2.SYSRSn
3	IPC3	R	0h	0: No IPC3 event request to CPU2 1: IPC3 event request to CPU2 Reset type: CPU2.SYSRSn
2	IPC2	R	0h	0: No IPC2 event request to CPU2 1: IPC2 event request to CPU2 Reset type: CPU2.SYSRSn
1	IPC1	R	0h	0: No IPC1 event request to CPU2 1: IPC1 event request to CPU2 Reset type: CPU2.SYSRSn
0	IPC0	R	0h	0: No IPC0 event request to CPU2 1: IPC0 event request to CPU2 Reset type: CPU2.SYSRSn

16.7.5.11 CPU3TOCPU2INTIPCSENDCOM_j Register (Offset = 2010h + formula) [Reset = 00000000h]

CPU3TOCPU2INTIPCSENDCOM_j is shown in Figure 16-48 and described in Table 16-58.

Return to the Summary Table.

CPU3 to CPU2 IPC Command

Offset = 2010h + (j * 800h); where j = 0h to 3h

Figure 16-48. CPU3TOCPU2INTIPCSENDCOM_j Register

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

Table 16-58. CPU3TOCPU2INTIPCSENDCOM_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMMAND	R/W	0h	This is a general purpose register used to send software-defined commands to from CPU3 to CPU2 Reset type: CPUx.SYSRSn

16.7.5.12 CPU3TOCPU2INTIPCSENDADDR_j Register (Offset = 2014h + formula) [Reset = 00000000h]

CPU3TOCPU2INTIPCSENDADDR_j is shown in Figure 16-49 and described in Table 16-59.

Return to the Summary Table.

CPU3 to CPU2 IPC Address

Offset = 2014h + (j * 800h); where j = 0h to 3h

Figure 16-49. CPU3TOCPU2INTIPCSENDADDR_j Register

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

Table 16-59. CPU3TOCPU2INTIPCSENDADDR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R/W	0h	This is a general purpose register used to send software-defined Address to from CPU3 to CPU2 Reset type: CPUx.SYSRSn

16.7.5.13 CPU3TOCPU2INTIPCSENDDATA_j Register (Offset = 2018h + formula) [Reset = 00000000h]

CPU3TOCPU2INTIPCSENDDATA_j is shown in Figure 16-50 and described in Table 16-60.

Return to the Summary Table.

CPU3 to CPU2 IPC Data

Offset = 2018h + (j * 800h); where j = 0h to 3h

Figure 16-50. CPU3TOCPU2INTIPCSENDDATA_j Register

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

Table 16-60. CPU3TOCPU2INTIPCSENDDATA_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/W	0h	This is a general purpose register used to send software-defined Data to from CPU3 to CPU2 Reset type: CPUx.SYSRSn

16.7.5.14 CPU2TOCPU3INTREMOTEREPLY_j Register (Offset = 201Ch + formula) [Reset = 00000000h]

CPU2TOCPU3INTREMOTEREPLY_j is shown in Figure 16-51 and described in Table 16-61.

Return to the Summary Table.

Reply from CPU2 to CPU3TOCPU2IPCSENDCOM command request

Offset = 201Ch + (j * 800h); where j = 0h to 3h

Figure 16-51. CPU2TOCPU3INTREMOTEREPLY_j Register

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

Table 16-61. CPU2TOCPU3INTREMOTEREPLY_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REPLY	R	0h	Refelects the state of CPU2TOCPU3INT IPCREPLY register Reset type: CPUx.SYSRSn

16.7.5.15 CPU3TOHSMINTIPCSET_j Register (Offset = 6000h + formula) [Reset = 00000000h]

CPU3TOHSMINTIPCSET_j is shown in Figure 16-52 and described in Table 16-62.

Return to the Summary Table.

Set CPU3TOHSMINTIPCSET Register

Offset = 6000h + (j * 1000h); where j = 0h to 1h

Figure 16-52. CPU3TOHSMINTIPCSET_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-62. CPU3TOHSMINTIPCSET_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit sets the IPC31 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit sets the IPC30 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit sets the IPC29 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit sets the IPC28 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit sets the IPC27 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit sets the IPC26 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit sets the IPC25 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
24	IPC24	R-0/W1S	0h	Writing 1 to this bit sets the IPC24 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit sets the IPC23 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn

Table 16-62. CPU3TOHSMINTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R-0/W1S	0h	Writing 1 to this bit sets the IPC22 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit sets the IPC21 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit sets the IPC20 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit sets the IPC19 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit sets the IPC18 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit sets the IPC17 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit sets the IPC16 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit sets the IPC15 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit sets the IPC14 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit sets the IPC13 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit sets the IPC12 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit sets the IPC11 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
10	IPC10	R-0/W1S	0h	Writing 1 to this bit sets the IPC10 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit sets the IPC9 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit sets the IPC8 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit sets the IPC7 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit sets the IPC6 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn

Table 16-62. CPU3TOHSMINTIPCSET_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R-0/W1S	0h	Writing 1 to this bit sets the IPC5 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit sets the IPC4 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit sets the IPC3 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit sets the IPC2 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit sets the IPC1 event flag for the remote CPU. Writing 0 has no effect. Reset type: CPUx.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit sets the IPC0 event flag for the remote CPU. Writing 0 has no effect. Notes: [1] IPC event flags 0 will trigger interrupts in the receiving CPU via the PIPE. Reset type: CPUx.SYSRSn

16.7.5.16 CPU3TOHSMINTIPCCLR_j Register (Offset = 6004h + formula) [Reset = 00000000h]

CPU3TOHSMINTIPCCLR_j is shown in Figure 16-53 and described in Table 16-63.

Return to the Summary Table.

Clear CPU3TOHSMIPCFLG register

Offset = 6004h + (j * 1000h); where j = 0h to 1h

Figure 16-53. CPU3TOHSMINTIPCCLR_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-63. CPU3TOHSMINTIPCCLR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC31 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC30 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC29 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC28 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC27 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC26 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC25 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn

Table 16-63. CPU3TOHSMINTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	IPC24	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC24 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC23 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
22	IPC22	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC22 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC21 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC20 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC19 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
18	IPC18	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC18 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC17 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC16 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC15 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC14 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC13 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC12 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC11 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn

Table 16-63. CPU3TOHSMINTIPCCLR_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	IPC10	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC10 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC9 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC8 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC7 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC6 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
5	IPC5	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC5 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC4 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC3 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC2 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC1 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit clear the CPU3TOHSMIPCFLG.IPC0 event flag for HSM. Writing 0 has no effect. Reset type: CPUx.SYSRSn

16.7.5.17 CPU3TOHSMINTIPCFLG_j Register (Offset = 6008h + formula) [Reset = 00000000h]

CPU3TOHSMINTIPCFLG_j is shown in Figure 16-54 and described in Table 16-64.

Return to the Summary Table.

CPU3TOHSMINTIPCFLG Register

Offset = 6008h + (j * 1000h); where j = 0h to 1h

Figure 16-54. CPU3TOHSMINTIPCFLG_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-64. CPU3TOHSMINTIPCFLG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R	0h	0: No IPC31 event request to CPU3 1: IPC31 event request to CPU3 Reset type: CPUx.SYSRSn
30	IPC30	R	0h	0: No IPC30 event request to CPU3 1: IPC30 event request to CPU3 Reset type: CPUx.SYSRSn
29	IPC29	R	0h	0: No IPC29 event request to CPU3 1: IPC29 event request to CPU3 Reset type: CPUx.SYSRSn
28	IPC28	R	0h	0: No IPC28 event request to CPU3 1: IPC28 event request to CPU3 Reset type: CPUx.SYSRSn
27	IPC27	R	0h	0: No IPC27 event request to CPU3 1: IPC27 event request to CPU3 Reset type: CPUx.SYSRSn
26	IPC26	R	0h	0: No IPC26 event request to CPU3 1: IPC26 event request to CPU3 Reset type: CPUx.SYSRSn
25	IPC25	R	0h	0: No IPC25 event request to CPU3 1: IPC25 event request to CPU3 Reset type: CPUx.SYSRSn
24	IPC24	R	0h	0: No IPC24 event request to CPU3 1: IPC24 event request to CPU3 Reset type: CPUx.SYSRSn
23	IPC23	R	0h	0: No IPC23 event request to CPU3 1: IPC23 event request to CPU3 Reset type: CPUx.SYSRSn

Table 16-64. CPU3TOHSMINTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	IPC22	R	0h	0: No IPC22 event request to CPU3 1: IPC22 event request to CPU3 Reset type: CPUx.SYSRSn
21	IPC21	R	0h	0: No IPC21 event request to CPU3 1: IPC21 event request to CPU3 Reset type: CPUx.SYSRSn
20	IPC20	R	0h	0: No IPC20 event request to CPU3 1: IPC20 event request to CPU3 Reset type: CPUx.SYSRSn
19	IPC19	R	0h	0: No IPC19 event request to CPU3 1: IPC19 event request to CPU3 Reset type: CPUx.SYSRSn
18	IPC18	R	0h	0: No IPC18 event request to CPU3 1: IPC18 event request to CPU3 Reset type: CPUx.SYSRSn
17	IPC17	R	0h	0: No IPC17 event request to CPU3 1: IPC17 event request to CPU3 Reset type: CPUx.SYSRSn
16	IPC16	R	0h	0: No IPC16 event request to CPU3 1: IPC16 event request to CPU3 Reset type: CPUx.SYSRSn
15	IPC15	R	0h	0: No IPC15 event request to CPU3 1: IPC15 event request to CPU3 Reset type: CPUx.SYSRSn
14	IPC14	R	0h	0: No IPC14 event request to CPU3 1: IPC14 event request to CPU3 Reset type: CPUx.SYSRSn
13	IPC13	R	0h	0: No IPC13 event request to CPU3 1: IPC13 event request to CPU3 Reset type: CPUx.SYSRSn
12	IPC12	R	0h	0: No IPC12 event request to CPU3 1: IPC12 event request to CPU3 Reset type: CPUx.SYSRSn
11	IPC11	R	0h	0: No IPC11 event request to CPU3 1: IPC11 event request to CPU3 Reset type: CPUx.SYSRSn
10	IPC10	R	0h	0: No IPC10 event request to CPU3 1: IPC10 event request to CPU3 Reset type: CPUx.SYSRSn
9	IPC9	R	0h	0: No IPC9 event request to CPU3 1: IPC9 event request to CPU3 Reset type: CPUx.SYSRSn
8	IPC8	R	0h	0: No IPC8 event request to CPU3 1: IPC8 event request to CPU3 Reset type: CPUx.SYSRSn
7	IPC7	R	0h	0: No IPC7 event request to CPU3 1: IPC7 event request to CPU3 Reset type: CPUx.SYSRSn
6	IPC6	R	0h	0: No IPC6 event request to CPU3 1: IPC6 event request to CPU3 Reset type: CPUx.SYSRSn

Table 16-64. CPU3TOHSMINTIPCFLG_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	IPC5	R	0h	0: No IPC5 event request to CPU3 1: IPC5 event request to CPU3 Reset type: CPUx.SYSRSn
4	IPC4	R	0h	0: No IPC4 event request to CPU3 1: IPC4 event request to CPU3 Reset type: CPUx.SYSRSn
3	IPC3	R	0h	0: No IPC3 event request to CPU3 1: IPC3 event request to CPU3 Reset type: CPUx.SYSRSn
2	IPC2	R	0h	0: No IPC2 event request to CPU3 1: IPC2 event request to CPU3 Reset type: CPUx.SYSRSn
1	IPC1	R	0h	0: No IPC1 event request to CPU3 1: IPC1 event request to CPU3 Reset type: CPUx.SYSRSn
0	IPC0	R	0h	0: No IPC0 event request to CPU3 1: IPC0 event request to CPU3 Reset type: CPUx.SYSRSn

16.7.6 CPU1_IPC_RCV_REGS Registers

Table 16-65 lists the memory-mapped registers for the CPU1_IPC_RCV_REGS registers. All register offset addresses not listed in Table 16-65 should be considered as reserved locations and the register contents should not be modified.

Table 16-65. CPU1_IPC_RCV_REGS Registers

Offset	Acronym	Register Name	Protection
0h + formula	CPU2TOCPU1INTIPCSTS_j	CPU2TOCPU1INTIPCSTS Register	
4h + formula	CPU1TOCPU2INTIPCACK_j	CPU1TOCPU2INTIPCACK Register	
10h + formula	CPU2TOCPU1INTIPCRECVCOM_j	CPU2TOCPU1INTIPCRECVCOM Register	
14h + formula	CPU2TOCPU1INTIPCRECVADDR_j	CPU2TOCPU1INTIPCRECVADDR Register	
18h + formula	CPU2TOCPU1INTIPCRECVDATA_j	CPU2TOCPU1INTIPCRECVDATA Register	
1Ch + formula	CPU1TOCPU2INTLOCALREPLY_j	CPU1TOCPU2INTLOCALREPLY Register	
2000h + formula	CPU3TOCPU1INTIPCSTS_j	CPU3TOCPU1INTIPCSTS Register	
2004h + formula	CPU1TOCPU3INTIPCACK_j	CPU1TOCPU3INTIPCACK Register	
2010h + formula	CPU3TOCPU1INTIPCRECVCOM_j	CPU3TOCPU1INTIPCRECVCOM Register	
2014h + formula	CPU3TOCPU1INTIPCRECVADDR_j	CPU3TOCPU1INTIPCRECVADDR Register	
2018h + formula	CPU3TOCPU1INTIPCRECVDATA_j	CPU3TOCPU1INTIPCRECVDATA Register	
201Ch + formula	CPU1TOCPU3INTLOCALREPLY_j	CPU1TOCPU3INTLOCALREPLY Register	

Complex bit access types are encoded to fit into small table cells. Table 16-66 shows the codes that are used for access types in this section.

Table 16-66. CPU1_IPC_RCV_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.

Table 16-66. CPU1_IPC_RCV_REGS Access Type Codes (continued)

Access Type	Code	Description
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

16.7.6.1 CPU2TOCPU1INTIPCSTS_j Register (Offset = 0h + formula) [Reset = 00000000h]

CPU2TOCPU1INTIPCSTS_j is shown in Figure 16-55 and described in Table 16-67.

Return to the Summary Table.

Status of CPU1TOCPU2IPCFLG register

Offset = 0h + (j * 800h); where j = 0h to 3h

Figure 16-55. CPU2TOCPU1INTIPCSTS_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-67. CPU2TOCPU1INTIPCSTS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R	0h	Indicates to the local CPU if the IPC31 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
30	IPC30	R	0h	Indicates to the local CPU if the IPC30 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
29	IPC29	R	0h	Indicates to the local CPU if the IPC29 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
28	IPC28	R	0h	Indicates to the local CPU if the IPC28 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
27	IPC27	R	0h	Indicates to the local CPU if the IPC27 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
26	IPC26	R	0h	Indicates to the local CPU if the IPC26 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn

Table 16-67. CPU2TOCPU1INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	IPC25	R	0h	Indicates to the local CPU if the IPC25 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
24	IPC24	R	0h	Indicates to the local CPU if the IPC24 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
23	IPC23	R	0h	Indicates to the local CPU if the IPC23 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
22	IPC22	R	0h	Indicates to the local CPU if the IPC22 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
21	IPC21	R	0h	Indicates to the local CPU if the IPC21 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
20	IPC20	R	0h	Indicates to the local CPU if the IPC20 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
19	IPC19	R	0h	Indicates to the local CPU if the IPC19 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
18	IPC18	R	0h	Indicates to the local CPU if the IPC18 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
17	IPC17	R	0h	Indicates to the local CPU if the IPC17 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
16	IPC16	R	0h	Indicates to the local CPU if the IPC16 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
15	IPC15	R	0h	Indicates to the local CPU if the IPC15 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn

Table 16-67. CPU2TOCPU1INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	IPC14	R	0h	Indicates to the local CPU if the IPC14 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
13	IPC13	R	0h	Indicates to the local CPU if the IPC13 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
12	IPC12	R	0h	Indicates to the local CPU if the IPC12 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
11	IPC11	R	0h	Indicates to the local CPU if the IPC11 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
10	IPC10	R	0h	Indicates to the local CPU if the IPC10 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
9	IPC9	R	0h	Indicates to the local CPU if the IPC9 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
8	IPC8	R	0h	Indicates to the local CPU if the IPC8 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
7	IPC7	R	0h	Indicates to the local CPU if the IPC7 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
6	IPC6	R	0h	Indicates to the local CPU if the IPC6 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
5	IPC5	R	0h	Indicates to the local CPU if the IPC5 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
4	IPC4	R	0h	Indicates to the local CPU if the IPC4 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn

Table 16-67. CPU2TOCPU1INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	IPC3	R	0h	Indicates to the local CPU if the IPC3 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
2	IPC2	R	0h	Indicates to the local CPU if the IPC2 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
1	IPC1	R	0h	Indicates to the local CPU if the IPC1 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
0	IPC0	R	0h	Indicates to the local CPU if the IPC0 event flag was set by the remote CPU. 0: No IPC0 event was set by the remote CPU 1: An IPC0 event was set by the remote CPU Notes [1] IPC event flags 0 will trigger interrupts in the receiving CPU via the PIPE. Reset type: CPU2.SYSRSn

16.7.6.2 CPU1TOCPU2INTIPCACK_j Register (Offset = 4h + formula) [Reset = 00000000h]

CPU1TOCPU2INTIPCACK_j is shown in Figure 16-56 and described in Table 16-68.

Return to the Summary Table.

CPU1TOCPU2INTIPCACK Register

Offset = 4h + (j * 800h); where j = 0h to 3h

Figure 16-56. CPU1TOCPU2INTIPCACK_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-68. CPU1TOCPU2INTIPCACK_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC31 bit. Reset type: CPU2.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC30 bit. Reset type: CPU2.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC29 bit. Reset type: CPU2.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC28 bit. Reset type: CPU2.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC27 bit. Reset type: CPU2.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC26 bit. Reset type: CPU2.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC25 bit. Reset type: CPU2.SYSRSn
24	IPC24	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC24 bit. Reset type: CPU2.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC23 bit. Reset type: CPU2.SYSRSn
22	IPC22	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC22 bit. Reset type: CPU2.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC21 bit. Reset type: CPU2.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC20 bit. Reset type: CPU2.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC19 bit. Reset type: CPU2.SYSRSn

Table 16-68. CPU1TOCPU2INTIPCACK_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	IPC18	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC18 bit. Reset type: CPU2.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC17 bit. Reset type: CPU2.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC16 bit. Reset type: CPU2.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC15 bit. Reset type: CPU2.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC14 bit. Reset type: CPU2.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC13 bit. Reset type: CPU2.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC12 bit. Reset type: CPU2.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC11 bit. Reset type: CPU2.SYSRSn
10	IPC10	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC10 bit. Reset type: CPU2.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC9 bit. Reset type: CPU2.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC8 bit. Reset type: CPU2.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC7 bit. Reset type: CPU2.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC6 bit. Reset type: CPU2.SYSRSn
5	IPC5	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC5 bit. Reset type: CPU2.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC4 bit. Reset type: CPU2.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC3 bit. Reset type: CPU2.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC2 bit. Reset type: CPU2.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC1 bit. Reset type: CPU2.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU1IPCFLG.IPC0 bit. Reset type: CPU2.SYSRSn

16.7.6.3 CPU2TOCPU1INTIPCRECVCOM_j Register (Offset = 10h + formula) [Reset = 00000000h]

CPU2TOCPU1INTIPCRECVCOM_j is shown in Figure 16-57 and described in Table 16-69.

Return to the Summary Table.

Refelects the value in CPU2TOCPU1IPCSENDCOM Register

Offset = 10h + (j * 800h); where j = 0h to 3h

Figure 16-57. CPU2TOCPU1INTIPCRECVCOM_j Register

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

Table 16-69. CPU2TOCPU1INTIPCRECVCOM_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMMAND	R	0h	Refelects the state of CPU2TOCPU1INT IPCRECVCOM register Reset type: CPU1.SYSRSn

16.7.6.4 CPU2TOCPU1INTIPCRECVADDR_j Register (Offset = 14h + formula) [Reset = 00000000h]

CPU2TOCPU1INTIPCRECVADDR_j is shown in Figure 16-58 and described in Table 16-70.

Return to the Summary Table.

Refelects the value in CPU2TOCPU1IPCSENDADDR Register

Offset = 14h + (j * 800h); where j = 0h to 3h

Figure 16-58. CPU2TOCPU1INTIPCRECVADDR_j Register

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

Table 16-70. CPU2TOCPU1INTIPCRECVADDR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Refelects the state of CPU2TOCPU1INT IPCRECVADDR register Reset type: CPU1.SYSRSn

16.7.6.5 CPU2TOCPU1INTIPCRECVDATA_j Register (Offset = 18h + formula) [Reset = 00000000h]

CPU2TOCPU1INTIPCRECVDATA_j is shown in Figure 16-59 and described in Table 16-71.

Return to the Summary Table.

Refelects the value in CPU2TOCPU1IPCSENDDATA Register

Offset = 18h + (j * 800h); where j = 0h to 3h

Figure 16-59. CPU2TOCPU1INTIPCRECVDATA_j Register

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

Table 16-71. CPU2TOCPU1INTIPCRECVDATA_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	Refelects the state of CPU2TOCPU1INT IPCRECVDATA register Reset type: CPU1.SYSRSn

16.7.6.6 CPU1TOCPU2INTLOCALREPLY_j Register (Offset = 1Ch + formula) [Reset = 00000000h]

CPU1TOCPU2INTLOCALREPLY_j is shown in Figure 16-60 and described in Table 16-72.

Return to the Summary Table.

Reply from CPU1 to CPU2TOCPU1IPCSENDCOM command

Offset = 1Ch + (j * 800h); where j = 0h to 3h

Figure 16-60. CPU1TOCPU2INTLOCALREPLY_j Register

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

Table 16-72. CPU1TOCPU2INTLOCALREPLY_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REPLY	R/W	0h	This is a general purpose register used to send software-defined REPLY to from CPU1 to CPU2 Reset type: CPU1.SYSRSn

16.7.6.7 CPU3TOCPU1INTIPCSTS_j Register (Offset = 2000h + formula) [Reset = 00000000h]

CPU3TOCPU1INTIPCSTS_j is shown in Figure 16-61 and described in Table 16-73.

Return to the Summary Table.

Status of CPU1TOCPU3IPCFLG register

Offset = 2000h + (j * 800h); where j = 0h to 3h

Figure 16-61. CPU3TOCPU1INTIPCSTS_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-73. CPU3TOCPU1INTIPCSTS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R	0h	Indicates to the local CPU if the IPC31 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
30	IPC30	R	0h	Indicates to the local CPU if the IPC30 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
29	IPC29	R	0h	Indicates to the local CPU if the IPC29 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
28	IPC28	R	0h	Indicates to the local CPU if the IPC28 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
27	IPC27	R	0h	Indicates to the local CPU if the IPC27 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
26	IPC26	R	0h	Indicates to the local CPU if the IPC26 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn

Table 16-73. CPU3TOCPU1INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	IPC25	R	0h	Indicates to the local CPU if the IPC25 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
24	IPC24	R	0h	Indicates to the local CPU if the IPC24 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
23	IPC23	R	0h	Indicates to the local CPU if the IPC23 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
22	IPC22	R	0h	Indicates to the local CPU if the IPC22 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
21	IPC21	R	0h	Indicates to the local CPU if the IPC21 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
20	IPC20	R	0h	Indicates to the local CPU if the IPC20 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
19	IPC19	R	0h	Indicates to the local CPU if the IPC19 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
18	IPC18	R	0h	Indicates to the local CPU if the IPC18 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
17	IPC17	R	0h	Indicates to the local CPU if the IPC17 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
16	IPC16	R	0h	Indicates to the local CPU if the IPC16 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
15	IPC15	R	0h	Indicates to the local CPU if the IPC15 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn

Table 16-73. CPU3TOCPU1INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	IPC14	R	0h	Indicates to the local CPU if the IPC14 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
13	IPC13	R	0h	Indicates to the local CPU if the IPC13 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
12	IPC12	R	0h	Indicates to the local CPU if the IPC12 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
11	IPC11	R	0h	Indicates to the local CPU if the IPC11 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
10	IPC10	R	0h	Indicates to the local CPU if the IPC10 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
9	IPC9	R	0h	Indicates to the local CPU if the IPC9 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
8	IPC8	R	0h	Indicates to the local CPU if the IPC8 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
7	IPC7	R	0h	Indicates to the local CPU if the IPC7 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
6	IPC6	R	0h	Indicates to the local CPU if the IPC6 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
5	IPC5	R	0h	Indicates to the local CPU if the IPC5 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
4	IPC4	R	0h	Indicates to the local CPU if the IPC4 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn

Table 16-73. CPU3TOCPU1INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	IPC3	R	0h	Indicates to the local CPU if the IPC3 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
2	IPC2	R	0h	Indicates to the local CPU if the IPC2 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
1	IPC1	R	0h	Indicates to the local CPU if the IPC1 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
0	IPC0	R	0h	Indicates to the local CPU if the IPC0 event flag was set by the remote CPU. 0: No IPC0 event was set by the remote CPU 1: An IPC0 event was set by the remote CPU Notes [1] IPC event flags 0 will trigger interrupts in the receiving CPU via the PIPE. Reset type: CPUx.SYSRSn

16.7.6.8 CPU1TOCPU3INTIPCACK_j Register (Offset = 2004h + formula) [Reset = 00000000h]

CPU1TOCPU3INTIPCACK_j is shown in Figure 16-62 and described in Table 16-74.

Return to the Summary Table.

CPU1TOCPU3INTIPCACK Register

Offset = 2004h + (j * 800h); where j = 0h to 3h

Figure 16-62. CPU1TOCPU3INTIPCACK_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-74. CPU1TOCPU3INTIPCACK_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC31 bit. Reset type: CPUx.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC30 bit. Reset type: CPUx.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC29 bit. Reset type: CPUx.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC28 bit. Reset type: CPUx.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC27 bit. Reset type: CPUx.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC26 bit. Reset type: CPUx.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC25 bit. Reset type: CPUx.SYSRSn
24	IPC24	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC24 bit. Reset type: CPUx.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC23 bit. Reset type: CPUx.SYSRSn
22	IPC22	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC22 bit. Reset type: CPUx.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC21 bit. Reset type: CPUx.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC20 bit. Reset type: CPUx.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC19 bit. Reset type: CPUx.SYSRSn

Table 16-74. CPU1TOCPU3INTIPCACK_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	IPC18	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC18 bit. Reset type: CPUx.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC17 bit. Reset type: CPUx.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC16 bit. Reset type: CPUx.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC15 bit. Reset type: CPUx.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC14 bit. Reset type: CPUx.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC13 bit. Reset type: CPUx.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC12 bit. Reset type: CPUx.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC11 bit. Reset type: CPUx.SYSRSn
10	IPC10	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC10 bit. Reset type: CPUx.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC9 bit. Reset type: CPUx.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC8 bit. Reset type: CPUx.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC7 bit. Reset type: CPUx.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC6 bit. Reset type: CPUx.SYSRSn
5	IPC5	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC5 bit. Reset type: CPUx.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC4 bit. Reset type: CPUx.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC3 bit. Reset type: CPUx.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC2 bit. Reset type: CPUx.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC1 bit. Reset type: CPUx.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU1IPCFLG.IPC0 bit. Reset type: CPUx.SYSRSn

16.7.6.9 CPU3TOCPU1INTIPCRECVCOM_j Register (Offset = 2010h + formula) [Reset = 00000000h]

CPU3TOCPU1INTIPCRECVCOM_j is shown in Figure 16-63 and described in Table 16-75.

Return to the Summary Table.

Refelects the value in CPU3TOCPU1IPCSENDCOM Register

Offset = 2010h + (j * 800h); where j = 0h to 3h

Figure 16-63. CPU3TOCPU1INTIPCRECVCOM_j Register

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

Table 16-75. CPU3TOCPU1INTIPCRECVCOM_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMMAND	R	0h	Refelects the state of CPU3TOCPU1INT IPCRECVCOM register Reset type: CPU1.SYSRSn

16.7.6.10 CPU3TOCPU1INTIPCRECVADDR_j Register (Offset = 2014h + formula) [Reset = 00000000h]

CPU3TOCPU1INTIPCRECVADDR_j is shown in Figure 16-64 and described in Table 16-76.

Return to the Summary Table.

Refelects the value in CPU3TOCPU1IPCSENDADDR Register

Offset = 2014h + (j * 800h); where j = 0h to 3h

Figure 16-64. CPU3TOCPU1INTIPCRECVADDR_j Register

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

Table 16-76. CPU3TOCPU1INTIPCRECVADDR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Refelects the state of CPU3TOCPU1INT IPCRECVADDR register Reset type: CPU1.SYSRSn

16.7.6.11 CPU3TOCPU1INTIPCRECVDATA_j Register (Offset = 2018h + formula) [Reset = 00000000h]

CPU3TOCPU1INTIPCRECVDATA_j is shown in Figure 16-65 and described in Table 16-77.

Return to the Summary Table.

Refelects the value in CPU3TOCPU1IPCSENDDATA Register

Offset = 2018h + (j * 800h); where j = 0h to 3h

Figure 16-65. CPU3TOCPU1INTIPCRECVDATA_j Register

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

Table 16-77. CPU3TOCPU1INTIPCRECVDATA_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	Refelects the state of CPU3TOCPU1INT IPCRECVDATA register Reset type: CPU1.SYSRSn

16.7.6.12 CPU1TOCPU3INTLOCALREPLY_j Register (Offset = 201Ch + formula) [Reset = 00000000h]

CPU1TOCPU3INTLOCALREPLY_j is shown in Figure 16-66 and described in Table 16-78.

Return to the Summary Table.

Reply from CPU1 to CPU3TOCPU1IPCSENDCOM command

Offset = 201Ch + (j * 800h); where j = 0h to 3h

Figure 16-66. CPU1TOCPU3INTLOCALREPLY_j Register

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

Table 16-78. CPU1TOCPU3INTLOCALREPLY_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REPLY	R/W	0h	This is a general purpose register used to send software-defined REPLY to from CPU1 to CPU3 Reset type: CPU1.SYSRSn

16.7.7 CPU2_IPC_RCV_REGS Registers

Table 16-79 lists the memory-mapped registers for the CPU2_IPC_RCV_REGS registers. All register offset addresses not listed in Table 16-79 should be considered as reserved locations and the register contents should not be modified.

Table 16-79. CPU2_IPC_RCV_REGS Registers

Offset	Acronym	Register Name	Protection
0h + formula	CPU1TOCPU2INTIPCSTS_j	CPU1TOCPU2INTIPCSTS Register	
4h + formula	CPU2TOCPU1INTIPCACK_j	CPU2TOCPU1INTIPCACK Register	
10h + formula	CPU1TOCPU2INTIPCRECVCOM_j	CPU1TOCPU2INTIPCRECVCOM Register	
14h + formula	CPU1TOCPU2INTIPCRECVADDR_j	CPU1TOCPU2INTIPCRECVADDR Register	
18h + formula	CPU1TOCPU2INTIPCRECVDATA_j	CPU1TOCPU2INTIPCRECVDATA Register	
1Ch + formula	CPU2TOCPU1INTLOCALREPLY_j	CPU2TOCPU1INTLOCALREPLY Register	
2000h + formula	CPU3TOCPU2INTIPCSTS_j	CPU3TOCPU2INTIPCSTS Register	
2004h + formula	CPU2TOCPU3INTIPCACK_j	CPU2TOCPU3INTIPCACK Register	
2010h + formula	CPU3TOCPU2INTIPCRECVCOM_j	CPU3TOCPU2INTIPCRECVCOM Register	
2014h + formula	CPU3TOCPU2INTIPCRECVADDR_j	CPU3TOCPU2INTIPCRECVADDR Register	
2018h + formula	CPU3TOCPU2INTIPCRECVDATA_j	CPU3TOCPU2INTIPCRECVDATA Register	
201Ch + formula	CPU2TOCPU3INTLOCALREPLY_j	CPU2TOCPU3INTLOCALREPLY Register	

Complex bit access types are encoded to fit into small table cells. Table 16-80 shows the codes that are used for access types in this section.

Table 16-80. CPU2_IPC_RCV_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.

Table 16-80. CPU2_IPC_RCV_REGS Access Type Codes (continued)

Access Type	Code	Description
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

16.7.7.1 CPU1TOCPU2INTIPCSTS_j Register (Offset = 0h + formula) [Reset = 00000000h]

CPU1TOCPU2INTIPCSTS_j is shown in Figure 16-67 and described in Table 16-81.

Return to the Summary Table.

Status of CPU2TOCPU1IPCFLG register

Offset = 0h + (j * 800h); where j = 0h to 3h

Figure 16-67. CPU1TOCPU2INTIPCSTS_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-81. CPU1TOCPU2INTIPCSTS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R	0h	Indicates to the local CPU if the IPC31 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
30	IPC30	R	0h	Indicates to the local CPU if the IPC30 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
29	IPC29	R	0h	Indicates to the local CPU if the IPC29 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
28	IPC28	R	0h	Indicates to the local CPU if the IPC28 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
27	IPC27	R	0h	Indicates to the local CPU if the IPC27 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
26	IPC26	R	0h	Indicates to the local CPU if the IPC26 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn

Table 16-81. CPU1TOCPU2INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	IPC25	R	0h	Indicates to the local CPU if the IPC25 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
24	IPC24	R	0h	Indicates to the local CPU if the IPC24 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
23	IPC23	R	0h	Indicates to the local CPU if the IPC23 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
22	IPC22	R	0h	Indicates to the local CPU if the IPC22 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
21	IPC21	R	0h	Indicates to the local CPU if the IPC21 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
20	IPC20	R	0h	Indicates to the local CPU if the IPC20 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
19	IPC19	R	0h	Indicates to the local CPU if the IPC19 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
18	IPC18	R	0h	Indicates to the local CPU if the IPC18 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
17	IPC17	R	0h	Indicates to the local CPU if the IPC17 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
16	IPC16	R	0h	Indicates to the local CPU if the IPC16 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
15	IPC15	R	0h	Indicates to the local CPU if the IPC15 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn

Table 16-81. CPU1TOCPU2INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	IPC14	R	0h	Indicates to the local CPU if the IPC14 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
13	IPC13	R	0h	Indicates to the local CPU if the IPC13 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
12	IPC12	R	0h	Indicates to the local CPU if the IPC12 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
11	IPC11	R	0h	Indicates to the local CPU if the IPC11 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
10	IPC10	R	0h	Indicates to the local CPU if the IPC10 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
9	IPC9	R	0h	Indicates to the local CPU if the IPC9 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
8	IPC8	R	0h	Indicates to the local CPU if the IPC8 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
7	IPC7	R	0h	Indicates to the local CPU if the IPC7 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
6	IPC6	R	0h	Indicates to the local CPU if the IPC6 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
5	IPC5	R	0h	Indicates to the local CPU if the IPC5 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
4	IPC4	R	0h	Indicates to the local CPU if the IPC4 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn

Table 16-81. CPU1TOCPU2INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	IPC3	R	0h	Indicates to the local CPU if the IPC3 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
2	IPC2	R	0h	Indicates to the local CPU if the IPC2 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
1	IPC1	R	0h	Indicates to the local CPU if the IPC1 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
0	IPC0	R	0h	Indicates to the local CPU if the IPC0 event flag was set by the remote CPU. 0: No IPC0 event was set by the remote CPU 1: An IPC0 event was set by the remote CPU Notes [1] IPC event flags 0 will trigger interrupts in the receiving CPU via the PIPE. Reset type: CPU1.SYSRSn

16.7.7.2 CPU2TOCPU1INTIPCACK_j Register (Offset = 4h + formula) [Reset = 00000000h]

CPU2TOCPU1INTIPCACK_j is shown in Figure 16-68 and described in Table 16-82.

Return to the Summary Table.

CPU2TOCPU1INTIPCACK Register

Offset = 4h + (j * 800h); where j = 0h to 3h

Figure 16-68. CPU2TOCPU1INTIPCACK_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-82. CPU2TOCPU1INTIPCACK_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC31 bit. Reset type: CPU1.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC30 bit. Reset type: CPU1.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC29 bit. Reset type: CPU1.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC28 bit. Reset type: CPU1.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC27 bit. Reset type: CPU1.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC26 bit. Reset type: CPU1.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC25 bit. Reset type: CPU1.SYSRSn
24	IPC24	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC24 bit. Reset type: CPU1.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC23 bit. Reset type: CPU1.SYSRSn
22	IPC22	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC22 bit. Reset type: CPU1.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC21 bit. Reset type: CPU1.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC20 bit. Reset type: CPU1.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC19 bit. Reset type: CPU1.SYSRSn

Table 16-82. CPU2TOCPU1INTIPCACK_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	IPC18	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC18 bit. Reset type: CPU1.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC17 bit. Reset type: CPU1.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC16 bit. Reset type: CPU1.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC15 bit. Reset type: CPU1.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC14 bit. Reset type: CPU1.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC13 bit. Reset type: CPU1.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC12 bit. Reset type: CPU1.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC11 bit. Reset type: CPU1.SYSRSn
10	IPC10	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC10 bit. Reset type: CPU1.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC9 bit. Reset type: CPU1.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC8 bit. Reset type: CPU1.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC7 bit. Reset type: CPU1.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC6 bit. Reset type: CPU1.SYSRSn
5	IPC5	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC5 bit. Reset type: CPU1.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC4 bit. Reset type: CPU1.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC3 bit. Reset type: CPU1.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC2 bit. Reset type: CPU1.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC1 bit. Reset type: CPU1.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU2IPCFLG.IPC0 bit. Reset type: CPU1.SYSRSn

16.7.7.3 CPU1TOCPU2INTIPCRECVCOM_j Register (Offset = 10h + formula) [Reset = 00000000h]

CPU1TOCPU2INTIPCRECVCOM_j is shown in Figure 16-69 and described in Table 16-83.

Return to the Summary Table.

Refelects the value in CPU1TOCPU2IPCSENDCOM Register

Offset = 10h + (j * 800h); where j = 0h to 3h

Figure 16-69. CPU1TOCPU2INTIPCRECVCOM_j Register

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

Table 16-83. CPU1TOCPU2INTIPCRECVCOM_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMMAND	R	0h	Refelects the state of CPU1TOCPU2INT IPCRECVCOM register Reset type: CPU2.SYSRSn

16.7.7.4 CPU1TOCPU2INTIPCRECVADDR_j Register (Offset = 14h + formula) [Reset = 00000000h]

CPU1TOCPU2INTIPCRECVADDR_j is shown in Figure 16-70 and described in Table 16-84.

Return to the Summary Table.

Refelects the value in CPU1TOCPU2IPCSENDADDR Register

Offset = 14h + (j * 800h); where j = 0h to 3h

Figure 16-70. CPU1TOCPU2INTIPCRECVADDR_j Register

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

Table 16-84. CPU1TOCPU2INTIPCRECVADDR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Refelects the state of CPU1TOCPU2INT IPCRECVADDR register Reset type: CPU2.SYSRSn

16.7.7.5 CPU1TOCPU2INTIPCRECVDATA_j Register (Offset = 18h + formula) [Reset = 00000000h]

CPU1TOCPU2INTIPCRECVDATA_j is shown in Figure 16-71 and described in Table 16-85.

Return to the Summary Table.

Refelects the value in CPU1TOCPU2IPCSENDDATA Register

Offset = 18h + (j * 800h); where j = 0h to 3h

Figure 16-71. CPU1TOCPU2INTIPCRECVDATA_j Register

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

Table 16-85. CPU1TOCPU2INTIPCRECVDATA_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	Refelects the state of CPU1TOCPU2INT IPCRECVDATA register Reset type: CPU2.SYSRSn

16.7.7.6 CPU2TOCPU1INTLOCALREPLY_j Register (Offset = 1Ch + formula) [Reset = 00000000h]

CPU2TOCPU1INTLOCALREPLY_j is shown in Figure 16-72 and described in Table 16-86.

Return to the Summary Table.

Reply from CPU2 to CPU1TOCPU2IPCSENDCOM command

Offset = 1Ch + (j * 800h); where j = 0h to 3h

Figure 16-72. CPU2TOCPU1INTLOCALREPLY_j Register

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

Table 16-86. CPU2TOCPU1INTLOCALREPLY_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REPLY	R/W	0h	This is a general purpose register used to send software-defined REPLY to from CPU2 to CPU1 Reset type: CPU2.SYSRSn

16.7.7.7 CPU3TOCPU2INTIPCSTS_j Register (Offset = 2000h + formula) [Reset = 00000000h]

CPU3TOCPU2INTIPCSTS_j is shown in Figure 16-73 and described in Table 16-87.

Return to the Summary Table.

Status of CPU2TOCPU3IPCFLG register

Offset = 2000h + (j * 800h); where j = 0h to 3h

Figure 16-73. CPU3TOCPU2INTIPCSTS_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-87. CPU3TOCPU2INTIPCSTS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R	0h	Indicates to the local CPU if the IPC31 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
30	IPC30	R	0h	Indicates to the local CPU if the IPC30 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
29	IPC29	R	0h	Indicates to the local CPU if the IPC29 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
28	IPC28	R	0h	Indicates to the local CPU if the IPC28 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
27	IPC27	R	0h	Indicates to the local CPU if the IPC27 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
26	IPC26	R	0h	Indicates to the local CPU if the IPC26 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn

Table 16-87. CPU3TOCPU2INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	IPC25	R	0h	Indicates to the local CPU if the IPC25 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
24	IPC24	R	0h	Indicates to the local CPU if the IPC24 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
23	IPC23	R	0h	Indicates to the local CPU if the IPC23 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
22	IPC22	R	0h	Indicates to the local CPU if the IPC22 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
21	IPC21	R	0h	Indicates to the local CPU if the IPC21 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
20	IPC20	R	0h	Indicates to the local CPU if the IPC20 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
19	IPC19	R	0h	Indicates to the local CPU if the IPC19 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
18	IPC18	R	0h	Indicates to the local CPU if the IPC18 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
17	IPC17	R	0h	Indicates to the local CPU if the IPC17 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
16	IPC16	R	0h	Indicates to the local CPU if the IPC16 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
15	IPC15	R	0h	Indicates to the local CPU if the IPC15 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn

Table 16-87. CPU3TOCPU2INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	IPC14	R	0h	Indicates to the local CPU if the IPC14 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
13	IPC13	R	0h	Indicates to the local CPU if the IPC13 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
12	IPC12	R	0h	Indicates to the local CPU if the IPC12 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
11	IPC11	R	0h	Indicates to the local CPU if the IPC11 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
10	IPC10	R	0h	Indicates to the local CPU if the IPC10 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
9	IPC9	R	0h	Indicates to the local CPU if the IPC9 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
8	IPC8	R	0h	Indicates to the local CPU if the IPC8 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
7	IPC7	R	0h	Indicates to the local CPU if the IPC7 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
6	IPC6	R	0h	Indicates to the local CPU if the IPC6 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
5	IPC5	R	0h	Indicates to the local CPU if the IPC5 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
4	IPC4	R	0h	Indicates to the local CPU if the IPC4 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn

Table 16-87. CPU3TOCPU2INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	IPC3	R	0h	Indicates to the local CPU if the IPC3 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
2	IPC2	R	0h	Indicates to the local CPU if the IPC2 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
1	IPC1	R	0h	Indicates to the local CPU if the IPC1 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPUx.SYSRSn
0	IPC0	R	0h	Indicates to the local CPU if the IPC0 event flag was set by the remote CPU. 0: No IPC0 event was set by the remote CPU 1: An IPC0 event was set by the remote CPU Notes [1] IPC event flags 0 will trigger interrupts in the receiving CPU via the PIPE. Reset type: CPUx.SYSRSn

16.7.7.8 CPU2TOCPU3INTIPCACK_j Register (Offset = 2004h + formula) [Reset = 00000000h]

CPU2TOCPU3INTIPCACK_j is shown in Figure 16-74 and described in Table 16-88.

Return to the Summary Table.

CPU2TOCPU3INTIPCACK Register

Offset = 2004h + (j * 800h); where j = 0h to 3h

Figure 16-74. CPU2TOCPU3INTIPCACK_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-88. CPU2TOCPU3INTIPCACK_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC31 bit. Reset type: CPUx.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC30 bit. Reset type: CPUx.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC29 bit. Reset type: CPUx.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC28 bit. Reset type: CPUx.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC27 bit. Reset type: CPUx.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC26 bit. Reset type: CPUx.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC25 bit. Reset type: CPUx.SYSRSn
24	IPC24	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC24 bit. Reset type: CPUx.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC23 bit. Reset type: CPUx.SYSRSn
22	IPC22	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC22 bit. Reset type: CPUx.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC21 bit. Reset type: CPUx.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC20 bit. Reset type: CPUx.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC19 bit. Reset type: CPUx.SYSRSn

Table 16-88. CPU2TOCPU3INTIPCACK_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	IPC18	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC18 bit. Reset type: CPUx.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC17 bit. Reset type: CPUx.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC16 bit. Reset type: CPUx.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC15 bit. Reset type: CPUx.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC14 bit. Reset type: CPUx.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC13 bit. Reset type: CPUx.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC12 bit. Reset type: CPUx.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC11 bit. Reset type: CPUx.SYSRSn
10	IPC10	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC10 bit. Reset type: CPUx.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC9 bit. Reset type: CPUx.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC8 bit. Reset type: CPUx.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC7 bit. Reset type: CPUx.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC6 bit. Reset type: CPUx.SYSRSn
5	IPC5	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC5 bit. Reset type: CPUx.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC4 bit. Reset type: CPUx.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC3 bit. Reset type: CPUx.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC2 bit. Reset type: CPUx.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC1 bit. Reset type: CPUx.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit, will clear CPU3TOCPU2IPCFLG.IPC0 bit. Reset type: CPUx.SYSRSn

16.7.7.9 CPU3TOCPU2INTIPCRECVCOM_j Register (Offset = 2010h + formula) [Reset = 00000000h]

CPU3TOCPU2INTIPCRECVCOM_j is shown in Figure 16-75 and described in Table 16-89.

Return to the Summary Table.

Refelects the value in CPU3TOCPU2IPCSENDCOM Register

Offset = 2010h + (j * 800h); where j = 0h to 3h

Figure 16-75. CPU3TOCPU2INTIPCRECVCOM_j Register

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

Table 16-89. CPU3TOCPU2INTIPCRECVCOM_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMMAND	R	0h	Refelects the state of CPU3TOCPU2INT IPCRECVCOM register Reset type: CPU2.SYSRSn

16.7.7.10 CPU3TOCPU2INTIPCRECVADDR_j Register (Offset = 2014h + formula) [Reset = 00000000h]

CPU3TOCPU2INTIPCRECVADDR_j is shown in Figure 16-76 and described in Table 16-90.

Return to the Summary Table.

Refelects the value in CPU3TOCPU2IPCSENDADDR Register

Offset = 2014h + (j * 800h); where j = 0h to 3h

Figure 16-76. CPU3TOCPU2INTIPCRECVADDR_j Register

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

Table 16-90. CPU3TOCPU2INTIPCRECVADDR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Refelects the state of CPU3TOCPU2INT IPCRECVADDR register Reset type: CPU2.SYSRSn

16.7.7.11 CPU3TOCPU2INTIPCRECVDATA_j Register (Offset = 2018h + formula) [Reset = 00000000h]

CPU3TOCPU2INTIPCRECVDATA_j is shown in Figure 16-77 and described in Table 16-91.

Return to the Summary Table.

Refelects the value in CPU3TOCPU2IPCSENDDATA Register

Offset = 2018h + (j * 800h); where j = 0h to 3h

Figure 16-77. CPU3TOCPU2INTIPCRECVDATA_j Register

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

Table 16-91. CPU3TOCPU2INTIPCRECVDATA_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	Refelects the state of CPU3TOCPU2INT IPCRECVDATA register Reset type: CPU2.SYSRSn

16.7.7.12 CPU2TOCPU3INTLOCALREPLY_j Register (Offset = 201Ch + formula) [Reset = 00000000h]

CPU2TOCPU3INTLOCALREPLY_j is shown in Figure 16-78 and described in Table 16-92.

Return to the Summary Table.

Reply from CPU2 to CPU3TOCPU2IPCSENDCOM command

Offset = 201Ch + (j * 800h); where j = 0h to 3h

Figure 16-78. CPU2TOCPU3INTLOCALREPLY_j Register

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

Table 16-92. CPU2TOCPU3INTLOCALREPLY_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REPLY	R/W	0h	This is a general purpose register used to send software-defined REPLY to from CPU2 to CPU3 Reset type: CPU2.SYSRSn

16.7.8 CPU3_IPC_RCV_REGS Registers

Table 16-93 lists the memory-mapped registers for the CPU3_IPC_RCV_REGS registers. All register offset addresses not listed in Table 16-93 should be considered as reserved locations and the register contents should not be modified.

Table 16-93. CPU3_IPC_RCV_REGS Registers

Offset	Acronym	Register Name	Protection
0h + formula	CPU1TOCPU3INTIPCSTS_j	CPU1TOCPU3INTIPCSTS Register	
4h + formula	CPU3TOCPU1INTIPCACK_j	CPU3TOCPU1INTIPCACK Register	
10h + formula	CPU1TOCPU3INTIPCRECVCOM_j	CPU1TOCPU3INTIPCRECVCOM Register	
14h + formula	CPU1TOCPU3INTIPCRECVADDR_j	CPU1TOCPU3INTIPCRECVADDR Register	
18h + formula	CPU1TOCPU3INTIPCRECVDATA_j	CPU1TOCPU3INTIPCRECVDATA Register	
1Ch + formula	CPU3TOCPU1INTLOCALREPLY_j	CPU3TOCPU1INTLOCALREPLY Register	
2000h + formula	CPU2TOCPU3INTIPCSTS_j	CPU2TOCPU3INTIPCSTS Register	
2004h + formula	CPU3TOCPU2INTIPCACK_j	CPU3TOCPU2INTIPCACK Register	
2010h + formula	CPU2TOCPU3INTIPCRECVCOM_j	CPU2TOCPU3INTIPCRECVCOM Register	
2014h + formula	CPU2TOCPU3INTIPCRECVADDR_j	CPU2TOCPU3INTIPCRECVADDR Register	
2018h + formula	CPU2TOCPU3INTIPCRECVDATA_j	CPU2TOCPU3INTIPCRECVDATA Register	
201Ch + formula	CPU3TOCPU2INTLOCALREPLY_j	CPU3TOCPU2INTLOCALREPLY Register	

Complex bit access types are encoded to fit into small table cells. Table 16-94 shows the codes that are used for access types in this section.

Table 16-94. CPU3_IPC_RCV_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.

Table 16-94. CPU3_IPC_RCV_REGS Access Type Codes (continued)

Access Type	Code	Description
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

16.7.8.1 CPU1TOCPU3INTIPCSTS_j Register (Offset = 0h + formula) [Reset = 00000000h]

CPU1TOCPU3INTIPCSTS_j is shown in Figure 16-79 and described in Table 16-95.

Return to the Summary Table.

Status of CPU3TOCPU1IPCFLG register

Offset = 0h + (j * 800h); where j = 0h to 3h

Figure 16-79. CPU1TOCPU3INTIPCSTS_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-95. CPU1TOCPU3INTIPCSTS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R	0h	Indicates to the local CPU if the IPC31 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
30	IPC30	R	0h	Indicates to the local CPU if the IPC30 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
29	IPC29	R	0h	Indicates to the local CPU if the IPC29 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
28	IPC28	R	0h	Indicates to the local CPU if the IPC28 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
27	IPC27	R	0h	Indicates to the local CPU if the IPC27 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
26	IPC26	R	0h	Indicates to the local CPU if the IPC26 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn

Table 16-95. CPU1TOCPU3INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	IPC25	R	0h	Indicates to the local CPU if the IPC25 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
24	IPC24	R	0h	Indicates to the local CPU if the IPC24 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
23	IPC23	R	0h	Indicates to the local CPU if the IPC23 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
22	IPC22	R	0h	Indicates to the local CPU if the IPC22 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
21	IPC21	R	0h	Indicates to the local CPU if the IPC21 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
20	IPC20	R	0h	Indicates to the local CPU if the IPC20 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
19	IPC19	R	0h	Indicates to the local CPU if the IPC19 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
18	IPC18	R	0h	Indicates to the local CPU if the IPC18 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
17	IPC17	R	0h	Indicates to the local CPU if the IPC17 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
16	IPC16	R	0h	Indicates to the local CPU if the IPC16 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
15	IPC15	R	0h	Indicates to the local CPU if the IPC15 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn

Table 16-95. CPU1TOCPU3INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	IPC14	R	0h	Indicates to the local CPU if the IPC14 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
13	IPC13	R	0h	Indicates to the local CPU if the IPC13 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
12	IPC12	R	0h	Indicates to the local CPU if the IPC12 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
11	IPC11	R	0h	Indicates to the local CPU if the IPC11 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
10	IPC10	R	0h	Indicates to the local CPU if the IPC10 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
9	IPC9	R	0h	Indicates to the local CPU if the IPC9 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
8	IPC8	R	0h	Indicates to the local CPU if the IPC8 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
7	IPC7	R	0h	Indicates to the local CPU if the IPC7 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
6	IPC6	R	0h	Indicates to the local CPU if the IPC6 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
5	IPC5	R	0h	Indicates to the local CPU if the IPC5 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
4	IPC4	R	0h	Indicates to the local CPU if the IPC4 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn

Table 16-95. CPU1TOCPU3INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	IPC3	R	0h	Indicates to the local CPU if the IPC3 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
2	IPC2	R	0h	Indicates to the local CPU if the IPC2 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
1	IPC1	R	0h	Indicates to the local CPU if the IPC1 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU1.SYSRSn
0	IPC0	R	0h	Indicates to the local CPU if the IPC0 event flag was set by the remote CPU. 0: No IPC0 event was set by the remote CPU 1: An IPC0 event was set by the remote CPU Notes [1] IPC event flags 0 will trigger interrupts in the receiving CPU via the PIPE. Reset type: CPU1.SYSRSn

16.7.8.2 CPU3TOCPU1INTIPCACK_j Register (Offset = 4h + formula) [Reset = 00000000h]

CPU3TOCPU1INTIPCACK_j is shown in Figure 16-80 and described in Table 16-96.

Return to the Summary Table.

CPU3TOCPU1INTIPCACK Register

Offset = 4h + (j * 800h); where j = 0h to 3h

Figure 16-80. CPU3TOCPU1INTIPCACK_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-96. CPU3TOCPU1INTIPCACK_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC31 bit. Reset type: CPU1.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC30 bit. Reset type: CPU1.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC29 bit. Reset type: CPU1.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC28 bit. Reset type: CPU1.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC27 bit. Reset type: CPU1.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC26 bit. Reset type: CPU1.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC25 bit. Reset type: CPU1.SYSRSn
24	IPC24	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC24 bit. Reset type: CPU1.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC23 bit. Reset type: CPU1.SYSRSn
22	IPC22	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC22 bit. Reset type: CPU1.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC21 bit. Reset type: CPU1.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC20 bit. Reset type: CPU1.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC19 bit. Reset type: CPU1.SYSRSn

Table 16-96. CPU3TOCPU1INTIPCACK_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	IPC18	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC18 bit. Reset type: CPU1.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC17 bit. Reset type: CPU1.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC16 bit. Reset type: CPU1.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC15 bit. Reset type: CPU1.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC14 bit. Reset type: CPU1.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC13 bit. Reset type: CPU1.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC12 bit. Reset type: CPU1.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC11 bit. Reset type: CPU1.SYSRSn
10	IPC10	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC10 bit. Reset type: CPU1.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC9 bit. Reset type: CPU1.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC8 bit. Reset type: CPU1.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC7 bit. Reset type: CPU1.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC6 bit. Reset type: CPU1.SYSRSn
5	IPC5	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC5 bit. Reset type: CPU1.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC4 bit. Reset type: CPU1.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC3 bit. Reset type: CPU1.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC2 bit. Reset type: CPU1.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC1 bit. Reset type: CPU1.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit, will clear CPU1TOCPU3IPCFLG.IPC0 bit. Reset type: CPU1.SYSRSn

16.7.8.3 CPU1TOCPU3INTIPCRECVCOM_j Register (Offset = 10h + formula) [Reset = 00000000h]

CPU1TOCPU3INTIPCRECVCOM_j is shown in Figure 16-81 and described in Table 16-97.

Return to the Summary Table.

Refelects the value in CPU1TOCPU3IPCSENDCOM Register

Offset = 10h + (j * 800h); where j = 0h to 3h

Figure 16-81. CPU1TOCPU3INTIPCRECVCOM_j Register

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

Table 16-97. CPU1TOCPU3INTIPCRECVCOM_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMMAND	R	0h	Refelects the state of CPU1TOCPU3INT IPCRECVCOM register Reset type: CPUx.SYSRSn

16.7.8.4 CPU1TOCPU3INTIPCRECVADDR_j Register (Offset = 14h + formula) [Reset = 00000000h]

CPU1TOCPU3INTIPCRECVADDR_j is shown in Figure 16-82 and described in Table 16-98.

Return to the Summary Table.

Refelects the value in CPU1TOCPU3IPCSENDADDR Register

Offset = 14h + (j * 800h); where j = 0h to 3h

Figure 16-82. CPU1TOCPU3INTIPCRECVADDR_j Register

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

Table 16-98. CPU1TOCPU3INTIPCRECVADDR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Refelects the state of CPU1TOCPU3INT IPCRECVADDR register Reset type: CPUx.SYSRSn

16.7.8.5 CPU1TOCPU3INTIPCRECVDATA_j Register (Offset = 18h + formula) [Reset = 00000000h]

CPU1TOCPU3INTIPCRECVDATA_j is shown in Figure 16-83 and described in Table 16-99.

Return to the Summary Table.

Refelects the value in CPU1TOCPU3IPCSENDDATA Register

Offset = 18h + (j * 800h); where j = 0h to 3h

Figure 16-83. CPU1TOCPU3INTIPCRECVDATA_j Register

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

Table 16-99. CPU1TOCPU3INTIPCRECVDATA_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	Refelects the state of CPU1TOCPU3INT IPCRECVDATA register Reset type: CPUx.SYSRSn

16.7.8.6 CPU3TOCPU1INTLOCALREPLY_j Register (Offset = 1Ch + formula) [Reset = 00000000h]

CPU3TOCPU1INTLOCALREPLY_j is shown in Figure 16-84 and described in Table 16-100.

Return to the Summary Table.

Reply from CPU3 to CPU1TOCPU3IPCSENDCOM command

Offset = 1Ch + (j * 800h); where j = 0h to 3h

Figure 16-84. CPU3TOCPU1INTLOCALREPLY_j Register

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

Table 16-100. CPU3TOCPU1INTLOCALREPLY_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REPLY	R/W	0h	This is a general purpose register used to send software-defined REPLY to from CPU3 to CPU1 Reset type: CPUx.SYSRSn

16.7.8.7 CPU2TOCPU3INTIPCSTS_j Register (Offset = 2000h + formula) [Reset = 00000000h]

CPU2TOCPU3INTIPCSTS_j is shown in Figure 16-85 and described in Table 16-101.

Return to the Summary Table.

Status of CPU3TOCPU2IPCFLG register

Offset = 2000h + (j * 800h); where j = 0h to 3h

Figure 16-85. CPU2TOCPU3INTIPCSTS_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-101. CPU2TOCPU3INTIPCSTS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R	0h	Indicates to the local CPU if the IPC31 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
30	IPC30	R	0h	Indicates to the local CPU if the IPC30 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
29	IPC29	R	0h	Indicates to the local CPU if the IPC29 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
28	IPC28	R	0h	Indicates to the local CPU if the IPC28 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
27	IPC27	R	0h	Indicates to the local CPU if the IPC27 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
26	IPC26	R	0h	Indicates to the local CPU if the IPC26 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn

Table 16-101. CPU2TOCPU3INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	IPC25	R	0h	Indicates to the local CPU if the IPC25 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
24	IPC24	R	0h	Indicates to the local CPU if the IPC24 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
23	IPC23	R	0h	Indicates to the local CPU if the IPC23 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
22	IPC22	R	0h	Indicates to the local CPU if the IPC22 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
21	IPC21	R	0h	Indicates to the local CPU if the IPC21 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
20	IPC20	R	0h	Indicates to the local CPU if the IPC20 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
19	IPC19	R	0h	Indicates to the local CPU if the IPC19 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
18	IPC18	R	0h	Indicates to the local CPU if the IPC18 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
17	IPC17	R	0h	Indicates to the local CPU if the IPC17 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
16	IPC16	R	0h	Indicates to the local CPU if the IPC16 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
15	IPC15	R	0h	Indicates to the local CPU if the IPC15 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn

Table 16-101. CPU2TOCPU3INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	IPC14	R	0h	Indicates to the local CPU if the IPC14 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
13	IPC13	R	0h	Indicates to the local CPU if the IPC13 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
12	IPC12	R	0h	Indicates to the local CPU if the IPC12 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
11	IPC11	R	0h	Indicates to the local CPU if the IPC11 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
10	IPC10	R	0h	Indicates to the local CPU if the IPC10 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
9	IPC9	R	0h	Indicates to the local CPU if the IPC9 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
8	IPC8	R	0h	Indicates to the local CPU if the IPC8 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
7	IPC7	R	0h	Indicates to the local CPU if the IPC7 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
6	IPC6	R	0h	Indicates to the local CPU if the IPC6 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
5	IPC5	R	0h	Indicates to the local CPU if the IPC5 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
4	IPC4	R	0h	Indicates to the local CPU if the IPC4 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn

Table 16-101. CPU2TOCPU3INTIPCSTS_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	IPC3	R	0h	Indicates to the local CPU if the IPC3 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
2	IPC2	R	0h	Indicates to the local CPU if the IPC2 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
1	IPC1	R	0h	Indicates to the local CPU if the IPC1 event flag was set by the remote CPU. 0: No IPC1 event was set by the remote CPU 1: An IPC1 event was set by the remote CPU Reset type: CPU2.SYSRSn
0	IPC0	R	0h	Indicates to the local CPU if the IPC0 event flag was set by the remote CPU. 0: No IPC0 event was set by the remote CPU 1: An IPC0 event was set by the remote CPU Notes [1] IPC event flags 0 will trigger interrupts in the receiving CPU via the PIPE. Reset type: CPU2.SYSRSn

16.7.8.8 CPU3TOCPU2INTIPCACK_j Register (Offset = 2004h + formula) [Reset = 00000000h]

CPU3TOCPU2INTIPCACK_j is shown in Figure 16-86 and described in Table 16-102.

Return to the Summary Table.

CPU3TOCPU2INTIPCACK Register

Offset = 2004h + (j * 800h); where j = 0h to 3h

Figure 16-86. CPU3TOCPU2INTIPCACK_j Register

31	30	29	28	27	26	25	24
IPC31	IPC30	IPC29	IPC28	IPC27	IPC26	IPC25	IPC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
IPC23	IPC22	IPC21	IPC20	IPC19	IPC18	IPC17	IPC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-102. CPU3TOCPU2INTIPCACK_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IPC31	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC31 bit. Reset type: CPU2.SYSRSn
30	IPC30	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC30 bit. Reset type: CPU2.SYSRSn
29	IPC29	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC29 bit. Reset type: CPU2.SYSRSn
28	IPC28	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC28 bit. Reset type: CPU2.SYSRSn
27	IPC27	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC27 bit. Reset type: CPU2.SYSRSn
26	IPC26	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC26 bit. Reset type: CPU2.SYSRSn
25	IPC25	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC25 bit. Reset type: CPU2.SYSRSn
24	IPC24	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC24 bit. Reset type: CPU2.SYSRSn
23	IPC23	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC23 bit. Reset type: CPU2.SYSRSn
22	IPC22	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC22 bit. Reset type: CPU2.SYSRSn
21	IPC21	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC21 bit. Reset type: CPU2.SYSRSn
20	IPC20	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC20 bit. Reset type: CPU2.SYSRSn
19	IPC19	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC19 bit. Reset type: CPU2.SYSRSn

Table 16-102. CPU3TOCPU2INTIPCACK_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	IPC18	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC18 bit. Reset type: CPU2.SYSRSn
17	IPC17	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC17 bit. Reset type: CPU2.SYSRSn
16	IPC16	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC16 bit. Reset type: CPU2.SYSRSn
15	IPC15	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC15 bit. Reset type: CPU2.SYSRSn
14	IPC14	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC14 bit. Reset type: CPU2.SYSRSn
13	IPC13	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC13 bit. Reset type: CPU2.SYSRSn
12	IPC12	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC12 bit. Reset type: CPU2.SYSRSn
11	IPC11	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC11 bit. Reset type: CPU2.SYSRSn
10	IPC10	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC10 bit. Reset type: CPU2.SYSRSn
9	IPC9	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC9 bit. Reset type: CPU2.SYSRSn
8	IPC8	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC8 bit. Reset type: CPU2.SYSRSn
7	IPC7	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC7 bit. Reset type: CPU2.SYSRSn
6	IPC6	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC6 bit. Reset type: CPU2.SYSRSn
5	IPC5	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC5 bit. Reset type: CPU2.SYSRSn
4	IPC4	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC4 bit. Reset type: CPU2.SYSRSn
3	IPC3	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC3 bit. Reset type: CPU2.SYSRSn
2	IPC2	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC2 bit. Reset type: CPU2.SYSRSn
1	IPC1	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC1 bit. Reset type: CPU2.SYSRSn
0	IPC0	R-0/W1S	0h	Writing 1 to this bit, will clear CPU2TOCPU3IPCFLG.IPC0 bit. Reset type: CPU2.SYSRSn

16.7.8.9 CPU2TOCPU3INTIPCRECVCOM_j Register (Offset = 2010h + formula) [Reset = 00000000h]

CPU2TOCPU3INTIPCRECVCOM_j is shown in Figure 16-87 and described in Table 16-103.

Return to the Summary Table.

Refelects the value in CPU2TOCPU3IPCSENDCOM Register

Offset = 2010h + (j * 800h); where j = 0h to 3h

Figure 16-87. CPU2TOCPU3INTIPCRECVCOM_j Register

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

Table 16-103. CPU2TOCPU3INTIPCRECVCOM_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMMAND	R	0h	Refelects the state of CPU2TOCPU3INT IPCRECVCOM register Reset type: CPUx.SYSRSn

16.7.8.10 CPU2TOCPU3INTIPCRECVADDR_j Register (Offset = 2014h + formula) [Reset = 00000000h]

CPU2TOCPU3INTIPCRECVADDR_j is shown in Figure 16-88 and described in Table 16-104.

Return to the Summary Table.

Refelects the value in CPU2TOCPU3IPCSENDADDR Register

Offset = 2014h + (j * 800h); where j = 0h to 3h

Figure 16-88. CPU2TOCPU3INTIPCRECVADDR_j Register

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

Table 16-104. CPU2TOCPU3INTIPCRECVADDR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Refelects the state of CPU2TOCPU3INT IPCRECVADDR register Reset type: CPUx.SYSRSn

16.7.8.11 CPU2TOCPU3INTIPCRECVDATA_j Register (Offset = 2018h + formula) [Reset = 00000000h]

CPU2TOCPU3INTIPCRECVDATA_j is shown in Figure 16-89 and described in Table 16-105.

Return to the Summary Table.

Refelects the value in CPU2TOCPU3IPCSENDDATA Register

Offset = 2018h + (j * 800h); where j = 0h to 3h

Figure 16-89. CPU2TOCPU3INTIPCRECVDATA_j Register

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

Table 16-105. CPU2TOCPU3INTIPCRECVDATA_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	Refelects the state of CPU2TOCPU3INT IPCRECVDATA register Reset type: CPUx.SYSRSn

16.7.8.12 CPU3TOCPU2INTLOCALREPLY_j Register (Offset = 201Ch + formula) [Reset = 00000000h]

CPU3TOCPU2INTLOCALREPLY_j is shown in Figure 16-90 and described in Table 16-106.

Return to the Summary Table.

Reply from CPU3 to CPU2TOCPU3IPCSENDCOM command

Offset = 201Ch + (j * 800h); where j = 0h to 3h

Figure 16-90. CPU3TOCPU2INTLOCALREPLY_j Register

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

Table 16-106. CPU3TOCPU2INTLOCALREPLY_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REPLY	R/W	0h	This is a general purpose register used to send software-defined REPLY to from CPU3 to CPU2 Reset type: CPUx.SYSRSn

Chapter 17

Embedded Real-time Analysis and Diagnostic (ERAD)

This chapter describes the features and operation of the embedded real-time analysis and diagnostic (ERAD) module. The ERAD module enhances the debug and system analysis capabilities of the device. The debug and system analysis enhancements provided by the ERAD module are implemented external to the CPU. The ERAD module consists of the enhanced bus comparator (EBC) units, and the system event counter (SEC) units. The EBC units are used to generate hardware breakpoints, hardware watch points, and other output events. The SEC units are used to analyze and profile the system. The ERAD module is accessible both by the debugger and the application software, which significantly increases the debug capabilities of many real-time systems, especially in situations where the debugger is not connected.

17.1 Introduction

The ERAD module is shown in Figure 17-1.

The ERAD enhances the debug and system analysis capabilities of the device external to the CPU. The ERAD module further expands this capability to provide additional hardware breakpoints, hardware watch points, and counters for profiling, as well as other advanced features. The ERAD module can be utilized by the debugger, and also by the application software. For many real-time systems, it is not always possible to connect a debugger and perform an intrusive debug. Under these situations, the user's code has the ability to set up and control the ERAD module to debug and profile the system without disturbing the end application.

Refer to the data sheet for the number of enhanced bus comparator (EBC) units and system event counter (SEC) units. The EBC units monitor buses and generate output events. The SEC units can be used with EBC units to profile and analyze the system. These units are described in detail in the following sections.

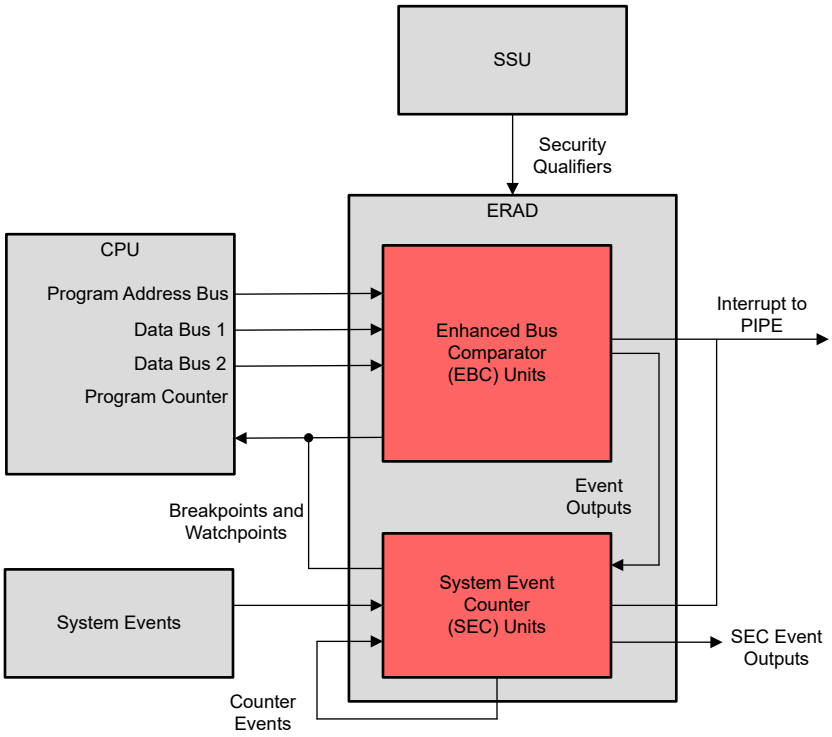


Figure 17-1. ERAD Overview

17.2 Enhanced Bus Comparator Unit

The Enhanced Bus Comparator (EBC) units connect to the CPU using a direct memory interface. This includes the program address and data buses, the data address, write and read data buses, debug qualifiers for memory access, and the ability to set breakpoints, watch points, and trace points on the CPU. Typically, the EBC is owned and controlled by the debugger application (for example, Code Composer Studio™ IDE). A user application running on the CPU can also configure and use the EBC units to generate events and interrupts for real-time diagnostic purposes. Note that ownership is exclusive—the debugger and application software cannot simultaneously control an EBC unit. For more information on EBC unit ownership, see Section 17.5.

The EBC units have the following capabilities:

- Generate hardware breakpoints
- Generate hardware watch points
- Generate trace tags for instruction fetch matches
- Monitor data address and read and write buses
- Generate event outputs that can be used by other modules.
- Generate interrupts and non-maskable interrupts
- Monitor program counter for currently executing instruction

The following features are not supported by the EBC units:

- Chained breakpoints
- Ability to monitor RTDMA transfers

Each EBC unit has the capability to monitor a range of addresses by defining masks and generating outputs based on greater than, less than, or equal events.

The EBC units can also be used with the System Event Counter (SEC) units for system or code profiling and analysis purposes.

17.2.1 Enhanced Bus Comparator Unit Operations

The following operations are supported by each EBC unit:

- Hardware Breakpoints: The EBC unit generates a break point tag when the specified instruction address is accessed on the program address bus. When the instruction reaches the DECODE-2 (D2) stage of the pipeline, the CPU is halted.
- Watch Points: A watch point detects a read or write to specified locations in data memory, and halts the CPU. Unlike hardware breakpoints, watch points do not have precise timing for halting the CPU—this is entirely dependent on the current state of the CPU pipeline. The CPU halts at the next interruptible boundary.
- Program Trace: Program traces are very similar to hardware breakpoints. The difference here is that instead of halting the CPU, a program trace generates an interrupt when the instruction reaches the D2 stage of the pipeline. If the instruction is discarded in the fetch buffer due to discontinuity, no interrupt is generated.
- Data Trace: A data trace is similar to a watch point, except that a data trace generates an interrupt instead of halting the CPU on an access to the specified data memory.

Note that hardware breakpoints only halt the CPU if a debugger is connected. If EBC or SEC is owned by application, no breakpoint or watch point can cause a halt on the CPU, even if EBC_CNTL.HALT is programmed to '1'.

17.2.2 Stack Qualification

Within the EBC, there is direct access to the stack pointer which can detect stack overflow scenarios. To configure the stack overflow detection:

1. Set register EBC_CNTL.STACK_QUAL = 1
2. Set register EBC_CNTL.BUS_SEL = 0x1
3. Set register EBC_CNTL.COMP_MODE = 0x4

17.2.3 Event Masking and Exporting

The events generated by different EBC units can be combined using OR and AND logic to generate new events as required, see Figure 17-2. There are AND and OR combinations that can be exported using masks to suppress undesired events. These events can be configured to generate an interrupt. The AND and OR events are also available as inputs to the system event counter unit for event counting and system profiling.

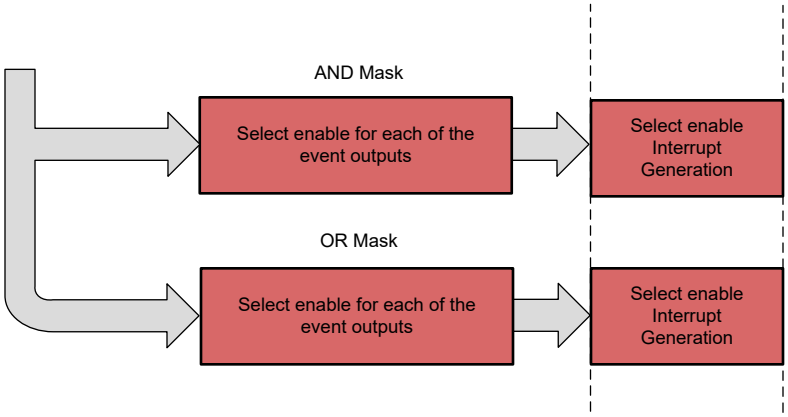


Figure 17-2. EBC Units Event Masking

To use the AND and OR masks:

1. Configure the EVENT_AND_MASK and OR_EVENT_MASK registers to select the desired EBC unit outputs for any of the available four masks.
2. To enable an interrupt for the configured mask, write 1 to the corresponding bit in the AND_MASK_CTL or OR_MASK_CTL register.
3. To use the mask output as an input to the system event counter unit, configure the SEC_INPUT_SEL1 register with the mux value for the desired mask. The input mux values are listed in Section 17.3.1.4.

For example, to generate a real-time interrupt on MASK1 when EBC units 2, 3, AND 6 events are triggered, write 0x26 to EVENT_AND_MASK, and then write 0x1 to AND_MASK_CTL[INTERRUPT].

17.3 System Event Counter Unit

The SEC units provide system profiling, analysis, and debug capability. The SEC units contain counters that can enhance the debug and profiling process in various types of system scenarios such as:

- Profiling code segments
- Counting duration between specified memory reads and writes
- Counting system events (such as interrupts)
- Counting duration between system events
- System timer
- Measuring the number of wait states in code segments
- Measuring the maximum amount of time spent in between a pair of events, measured over multiple iterations
- Chaining counters to link events or create larger counters

Furthermore, the SEC unit has the capability to:

- Function as a counter capable of counting:
  - Any of the match events generated by the EBC units.
  - Events generated by the EBC units. These events can be used to start and stop the counting.
  - System events including the interrupts to the interrupt controller, and timer interrupts. These system events can be used to start and stop the counting.
  - More information on the input sources for the SEC units can be found in Section 17.3.1.4.
- Generate an interrupt or a watch point if the count reaches a reference value.
- Perform counter operation in one of the following two modes:
  - Duration mode: The counter counts the CPU cycles as long as the event is active.
  - Event mode: The counter counts only the positive edge of the event signal. This is effectively counting the number of times the event transitions from inactive to active.

17.3.1 System Event Counter Modes

The following are the operating modes of the SEC unit. The counters are initialized to zero when the SEC module receives a reset input signal, and always count up.

- Continuous Count: In this mode, the counter continues to count as specified by the input selector. The counter can count the CPU cycles without any events selected. In this mode, the module can be used as a software-controlled SYSCLK counter. Continuous mode is active when SEC_CNTL.START_STOP_MODE and SEC_REF are both set to 0.
- Timer Mode Count: In this mode, the counter counts up to a set reference value, defined in the SEC_REF register. Upon reaching the reference value, the counter generates an event that can send an interrupt to the CPU or generate a watch point. The RST_ON_MATCH bit in the SEC_CNTL register configures the counter to either continue incrementing or reset when a match event occurs.
- Start-Stop Count: In this mode, two events are configured to act as start and stop indicators to the counter. The counter commences counting only when the defined start event occurs. The counter then continues to count up until the stop event occurs. Once the first start event has occurred, further start events are ignored until the stop event occurs.

In any of the counter modes of operation, there is a possibility that the 32-bit counter value overflows. If an overflow occurs, the counter value resets to zero and continues to count up, and the OVERFLOW bit in the SEC_STATUS register is set high. The OVERFLOW bit remains high until either the counter is reset, or the application writes 1 to the OVERFLOW bit of the SEC_STATUSCLEAR register.

17.3.1.1 Counting Active Levels Versus Edges

The SEC units can be configured to either count active levels or edges of the selected inputs.

Each SEC unit has eight inputs from the EBC units and many inputs from other events in the device. Each SEC unit can be configured to count any of the input events or just count up on every cycle. For example, if an input event occurs and is active for 25 cycles, the SEC unit counter increments only by 1 in event mode; whereas in the duration mode, the counter increments by 25.

17.3.1.2 Max and Min Mode

Max and min mode is also supported by the SEC units. This mode allows the user to detect the maximum and minimum count that has occurred during various count iterations in start-stop mode. For example, a user can set up the counter in the start-stop count mode to count the duration of a critical code loop. Every time the stop event occurs and the counter stops, the counter value is checked against the current MAX_COUNT and MIN_COUNT present in the register. If the new value is greater, then the MAX_COUNT register is updated. Similarly, if the current count is lesser than the MIN_COUNT, then the new value is loaded on to the MIN_COUNT register. The counter always resets to zero at the stop event and is ready to start counting on the next start event. Therefore, the MAX_COUNT contains the maximum number of cycles that occurred between the start and stop condition over many iterations. The MIN_COUNT register would show the minimum number of cycles.

17.3.1.3 Cumulative Mode

The SEC units can be used to yield the cumulative count over several start and stop events. In this mode the, unlike Max and Min mode, the counter does not reset due to a stop event. Instead it stops counting and resumes counting when a start event occurs. in cumulative count mode, the MAX_COUNT and MIN_COUNT is not valid.

17.3.1.4 Input Signal Selection

The SEC inputs can be selected from various signals from in the system to enable debug and system analysis. Figure 17-3 shows the SEC inputs. Each event selector MUX can select from various signals on in the system. These signals are shown in Table 17-1.

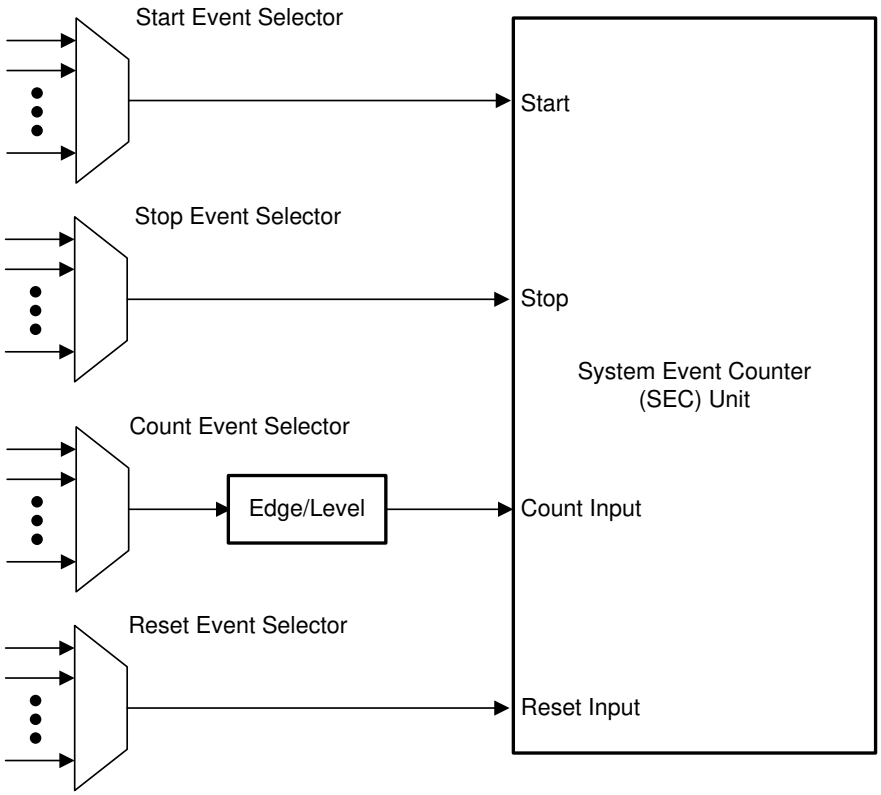


Figure 17-3. System Event Counter Inputs

Table 17-1. Event Selector Mux Signals

CTM\STA\STO\RST_INP_SEL	EVENT_INPUT_SELECTED	Polarity	Synchronization Requirement
0	EBC0	High	Disable
1	EBC1	High	Disable
2	EBC2	High	Disable
3	EBC3	High	Disable
4	EBC4	High	Disable
5	EBC5	High	Disable
6	EBC6	High	Disable
7	EBC7	High	Disable
8	COUNTER0_EVENT	High	Disable
9	COUNTER1_EVENT	High	Disable
10	COUNTER2_EVENT	High	Disable
11	COUNTER3_EVENT	High	Disable
12	ERAD_OR_MASK0	High	Disable
13	ERAD_OR_MASK1	High	Disable
14	ERAD_OR_MASK2	High	Disable
15	ERAD_OR_MASK3	High	Disable
16	ERAD_AND_MASK0	High	Disable
17	ERAD_AND_MASK1	High	Disable
18	ERAD_AND_MASK2	High	Disable
19	ERAD_AND_MASK3	High	Disable
20	PIPE_INT	High	Disable

Table 17-1. Event Selector Mux Signals (continued)

CTM\STA\STO\RST_INP_SEL	EVENT_INPUT_SELECTED	Polarity	Synchronization Requirement
21	PIPE_RTINT	High	Disable
22	PIPE_NMI	High	Disable
23	CPU1_TINT0	Low	Disable
24	CPU1_TINT1	Low	Disable
25	CPU1_TINT2	Low	Disable
26	CPU2_TINT0	Low	Disable
27	CPU2_TINT1	Low	Disable
28	CPU2_TINT2	Low	Disable
29	CPU3_TINT0	Low	Disable
30	CPU3_TINT1	Low	Disable
31	CPU3_TINT2	Low	Disable
32	RTDMA1_CH1INT	Low	Disable
33	RTDMA1_CH2INT	Low	Disable
34	RTDMA1_CH3INT	Low	Disable
35	RTDMA1_CH4INT	Low	Disable
36	RTDMA1_CH5INT	Low	Disable
37	RTDMA1_CH6INT	Low	Disable
38	RTDMA1_CH7INT	Low	Disable
39	RTDMA1_CH8INT	Low	Disable
40	RTDMA1_CH9INT	Low	Disable
41	RTDMA1_CH10INT	Low	Disable
42	ADCA_EVT_INT	Low	Disable
43	ADCB_EVT_INT	Low	Disable
44	ADCC_EVT_INT	Low	Disable
45	ADCD_EVT_INT	Low	Disable
46	ADCE_EVT_INT	Low	Disable
47	ADCSOCA	Low	Disable
48	ADCSOCB	Low	Disable
49	RTDMA2_CH1INT	Low	Disable
50	RTDMA2_CH2INT	Low	Disable
51	RTDMA2_CH3INT	Low	Disable
52	RTDMA2_CH4INT	Low	Disable
53	RTDMA2_CH5INT	Low	Disable
54	RTDMA2_CH6INT	Low	Disable
55	RTDMA2_CH7INT	Low	Disable
56	RTDMA2_CH8INT	Low	Disable
57	RTDMA2_CH9INT	Low	Disable
58	RTDMA2_CH10INT	Low	Disable
59	EPWMXBAR1	High	Disable
60	EPWMXBAR2	High	Disable
61	EPWMXBAR3	High	Disable
62	EPWMXBAR4	High	Disable
63	EPWMXBAR5	High	Disable
64	EPWMXBAR6	High	Disable

Table 17-1. Event Selector Mux Signals (continued)

CTM\STA\STO\RST_INP_SEL	EVENT_INPUT_SELECTED	Polarity	Synchronization Requirement
65	EPWMXBAR7	High	Disable
66	EPWMXBAR8	High	Disable
67	INPUTXBAR1	Low	Disable
68	INPUTXBAR2	Low	Disable
69	INPUTXBAR3	Low	Disable
70	INPUTXBAR4	Low	Disable
71	INPUTXBAR5	Low	Disable
72	INPUTXBAR6	Low	Disable
73	INPUTXBAR7	Low	Disable
74	INPUTXBAR8	Low	Disable
75	INPUTXBAR9	Low	Disable
76	INPUTXBAR10	Low	Disable
77	INPUTXBAR11	Low	Disable
78	INPUTXBAR12	Low	Disable
79	INPUTXBAR13	Low	Disable
80	INPUTXBAR14	Low	Disable
81	INPUTXBAR15	Low	Disable
82	INPUTXBAR16	Low	Disable
83	CMPSS1_CTRIPH_OR_CTRIPL	High	Disable
84	CMPSS2_CTRIPH_OR_CTRIPL	High	Disable
85	CMPSS3_CTRIPH_OR_CTRIPL	High	Disable
86	CMPSS4_CTRIPH_OR_CTRIPL	High	Disable
87	CMPSS5_CTRIPH_OR_CTRIPL	High	Disable
88	CMPSS6_CTRIPH_OR_CTRIPL	High	Disable
89	CMPSS7_CTRIPH_OR_CTRIPL	High	Disable
90	CMPSS8_CTRIPH_OR_CTRIPL	High	Disable
91	CMPSS9_CTRIPH_OR_CTRIPL	High	Disable
92	CMPSS10_CTRIPH_OR_CTRIPL	High	Disable
93	CMPSS11_CTRIPH_OR_CTRIPL	High	Disable
94	CMPSS12_CTRIPH_OR_CTRIPL	High	Disable
95	SD1FLT1_COMPH_OR_COMPL	High	Disable
96	SD1FLT2_COMPH_OR_COMPL	High	Disable
97	SD1FLT3_COMPH_OR_COMPL	High	Disable
98	SD1FLT4_COMPH_OR_COMPL	High	Disable
99	SD2FLT1_COMPH_OR_COMPL	High	Disable
100	SD2FLT2_COMPH_OR_COMPL	High	Disable
101	SD2FLT3_COMPH_OR_COMPL	High	Disable
102	SD2FLT4_COMPH_OR_COMPL	High	Disable
103	SD3FLT1_COMPH_OR_COMPL	High	Disable
104	SD3FLT2_COMPH_OR_COMPL	High	Disable
105	SD3FLT3_COMPH_OR_COMPL	High	Disable
106	SD3FLT4_COMPH_OR_COMPL	High	Disable
107	SD4FLT1_COMPH_OR_COMPL	High	Disable
108	SD4FLT2_COMPH_OR_COMPL	High	Disable

Table 17-1. Event Selector Mux Signals (continued)

CTM\STA\STO\RST_INP_SEL	EVENT_INPUT_SELECTED	Polarity	Synchronization Requirement
109	SD4FLT3_COMPH_OR_COMPL	High	Disable
110	SD4FLT4_COMPH_OR_COMPL	High	Disable
111	ADCAINT1	Low	Disable
112	ADCAINT2	Low	Disable
113	ADCAINT3	Low	Disable
114	ADCAINT4	Low	Disable
115	ADCBINT1	Low	Disable
116	ADCBINT2	Low	Disable
117	ADCBINT3	Low	Disable
118	ADCBINT4	Low	Disable
119	ADCCINT1	Low	Disable
120	ADCCINT2	Low	Disable
121	ADCCINT3	Low	Disable
122	ADCCINT4	Low	Disable
123	ADCDINT1	Low	Disable
124	ADCDINT2	Low	Disable
125	ADCDINT3	Low	Disable
126	ADCDINT4	Low	Disable
127	ADCEINT1	Low	Disable
128	ADCEINT2	Low	Disable
129	ADCEINT3	Low	Disable
130	ADCEINT4	Low	Disable
131	ECAT_PDI_SOF	High	Disable
132	ECAT_PDI_EOF	High	Disable
133	ECAT_PCI_WD_TRIGGER	High	Disable
134	ECAT_PDI_UC_IRQ	Low	Disable
135	ECAT_SYNCOUT0	High	Disable
136	ECAT_SYNCOUT1	High	Disable
137	ECAT_DRAM_PARITY_ERROR	High	Disable
138	INPUTXBAR17	Low	Disable
139	INPUTXBAR18	Low	Disable
140	INPUTXBAR19	Low	Disable
141	INPUTXBAR20	Low	Disable
142	INPUTXBAR21	Low	Disable
143	INPUTXBAR22	Low	Disable
144	INPUTXBAR23	Low	Disable
145	INPUTXBAR24	Low	Disable
146	INPUTXBAR25	Low	Disable
147	INPUTXBAR26	Low	Disable
148	INPUTXBAR27	Low	Disable
149	INPUTXBAR28	Low	Disable
150	INPUTXBAR29	Low	Disable
151	INPUTXBAR30	Low	Disable
152	INPUTXBAR31	Low	Disable

Table 17-1. Event Selector Mux Signals (continued)

CTM\STA\STO\RST_INP_SEL	EVENT_INPUT_SELECTED	Polarity	Synchronization Requirement
153	INPUTXBAR32	Low	Disable
154	FSIRXA_DATA_PKT_RCVD	High	Disable
155	FSIRXA_ERROR_PKT_RCVD	High	Disable
156	FSIRXA_PING_PKT_RCVD	High	Disable
157	FSIRXA_PING_TAG_MATCH	High	Disable
158	FSIRXA_DATA_TAG_MATCH	High	Disable
159	FSIRXA_ERROR_TAG_MATCH	High	Disable
160	FSIRXA_FRAME_DONE	High	Disable
161	FSIRXB_DATA_PKT_RCVD	High	Disable
162	FSIRXB_ERROR_PKT_RCVD	High	Disable
163	FSIRXB_PING_PKT_RCVD	High	Disable
164	FSIRXB_PING_TAG_MATCH	High	Disable
165	FSIRXB_DATA_TAG_MATCH	High	Disable
166	FSIRXB_ERROR_TAG_MATCH	High	Disable
167	FSIRXB_FRAME_DONE	High	Disable
168	FSIRXC_DATA_PKT_RCVD	High	Disable
169	FSIRXC_ERROR_PKT_RCVD	High	Disable
170	FSIRXC_PING_PKT_RCVD	High	Disable
171	FSIRXC_PING_TAG_MATCH	High	Disable
172	FSIRXC_DATA_TAG_MATCH	High	Disable
173	FSIRXC_ERROR_TAG_MATCH	High	Disable
174	FSIRXC_FRAME_DONE	High	Disable
175	FSIRXD_DATA_PKT_RCVD	High	Disable
176	FSIRXD_ERROR_PKT_RCVD	High	Disable
177	FSIRXD_PING_PKT_RCVD	High	Disable
178	FSIRXD_PING_TAG_MATCH	High	Disable
179	FSIRXD_DATA_TAG_MATCH	High	Disable
180	FSIRXD_ERROR_TAG_MATCH	High	Disable
181	FSIRXD_FRAME_DONE	High	Disable
182	TRACE_HIT_EVENT	High	Disable
183	CPU1_LCM_CMP_ERR	High	Disable
184	RTDMA_LCM_CMP_ERR	High	Disable
185	MCANA_EVT0	High	Disable
186	MCANA_EVT1	High	Disable
187	MCANA_EVT2	High	Disable
188	MCANB_EVT0	High	Disable
189	MCANB_EVT1	High	Disable
190	MCANB_EVT2	High	Disable
191	MCANC_EVT0	High	Disable
192	MCANC_EVT1	High	Disable
193	MCANC_EVT2	High	Disable
194	MCAND_EVT0	High	Disable
195	MCAND_EVT1	High	Disable
196	MCAND_EVT2	High	Disable

Table 17-1. Event Selector Mux Signals (continued)

CTM\STA\STO\RST_INP_SEL	EVENT_INPUT_SELECTED	Polarity	Synchronization Requirement
197	MCANE_EVT0	High	Disable
198	MCANE_EVT1	High	Disable
199	MCANE_EVT2	High	Disable
200	MCANF_EVT0	High	Disable
201	MCANF_EVT1	High	Disable
202	MCANF_EVT2	High	Disable
203	CPUx_int_ack	High	Disable
204	CPUx_rtint_ack	High	Disable
205	CPUx_vis_r1_pc_valid	High	Disable
206	CPUx_vis_w_dr1_req	High	Disable
207	CPUx_vis_w_dr2_req	High	Disable
208	CPUx_vis_w_dw_req	High	Disable
209	CPUx_cpi_d2_ready	Low	Disable
210	CPUx_cpi_r1_ready	Low	Disable
211	CPUx_cpi_exe_ready	Low	Disable
212-255	Reserved	Reserved	Reserved

17.3.2 Reset on Event

Resetting the counters on external events is also possible. Additionally all the counter event outputs are applied back to each of counter input MUX, which selects the event that can be used as a reset input. When enabled, an active high on the reset input causes the counter to reset. This gives a powerful feature that allows setting up threshold monitors. This can be used to flag an interrupt or a watch point, if the distance between two events crosses a certain threshold.

17.3.3 Operation Conditions

The SEC units count accurately only when the CPU is operating in normal conditions. If the counters are running and capturing the CPU cycles while the CPU is controlled through the debugger to single-step through the code, then the result can differ from when the CPU was executing the code in normal conditions.

17.4 Program Counter Trace

The Program Counter (PC) Trace module can be used to keep track of PC discontinuities or jumps, as a means to trace an entire program execution sequence over a period of time. When trace is completed or stopped, trace data can be read out using a debugger and analyzed to reconstruct the code execution sequence. The PC Trace module provides multiple modes and controls to govern when to trace and when not to trace. The trace module is tightly coupled with the Enhanced Bus Comparator, the System Event Counter Unit, and certain critical system level event signals.

Trace data is stored in an addressable memory buffer that can be read by software or a debugger. The trace data stored in this buffer includes additional status information on trace validity that can be used to correctly reconstruct the code execution system. For each discontinuity, two PC values are stored: the source of the discontinuity, and the destination. Figure 17-4 illustrates the operation of the PC trace module. The trace buffer is a circular buffer with overflow: when the buffer becomes full, new trace data is written starting from the top of the buffer, and an overflow status bit is set.

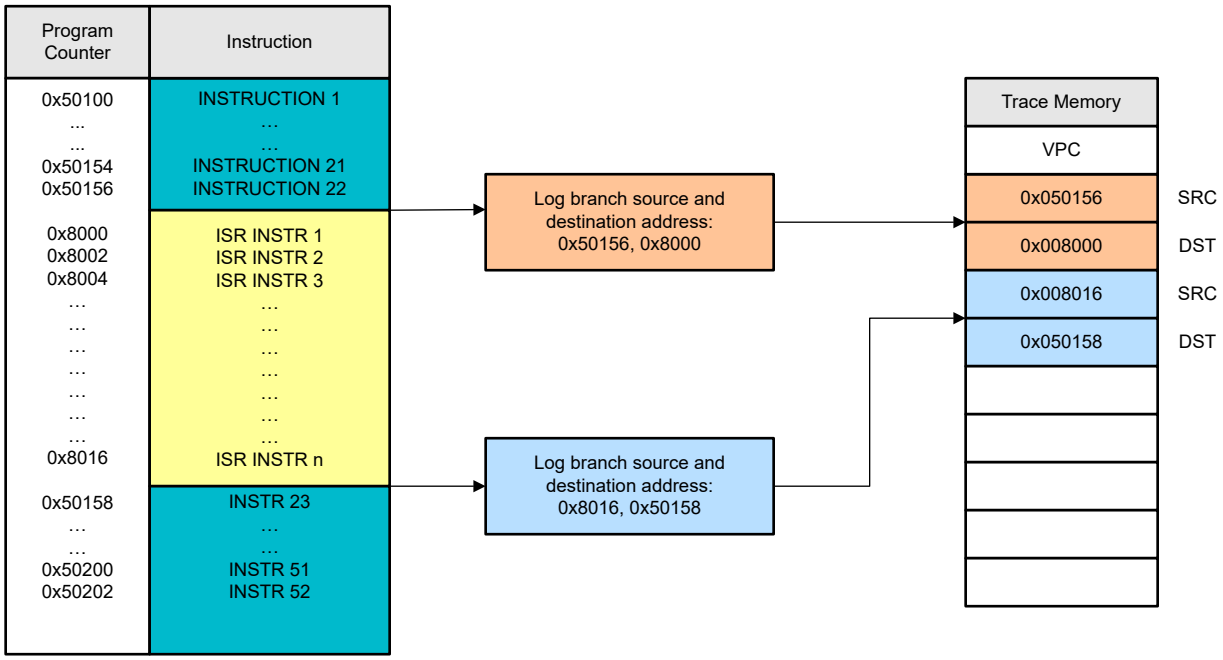


Figure 17-4. PC Trace Operation

17.4.1 Functional Block Diagram

Figure 17-5 describes the device PC trace architecture. The trace module is tightly coupled with the CPU, and receives the current program counter (VPC), program address (PAB), and various qualifying signals from the CPU interface. The Trace core captures these values whenever a PC discontinuity (for example, branch operation) is detected. The PC Trace module interfaces with the SSU. The SSU provides security sideband data needed to prevent unauthorized trace information from being exposed in trace memory. The PC Trace module interfaces with the Enhanced Bus Comparator Unit and System Event Counter Unit, providing the ability to select events from these units as triggers to start a trace, stop a trace, or determine the bounding conditions for a windowed trace operation.

The Trace Core qualifies trace source and destination addresses, and stores these addresses sequentially in the trace memory buffer. Additionally, the Trace Core can generate hit events every time a new trace is stored in the memory buffer; this event signal is connected to the ERAD counter block so that the number of entries in the buffer can be tracked. This counter value can in turn be used to create a STOP event at a predefined threshold.

Note

The Trace Core generates a trace hit event for a discontinuity arising from a speculative instruction fetch, even if the fetched instruction does not reach the execution phase in the CPU pipeline. As a result, the BUFFER_FULL signal can be set prematurely due to a speculative prefetch. The user can always safely discard the oldest discontinuity pair present in the memory buffer when a full buffer is detected, to mitigate this scenario.

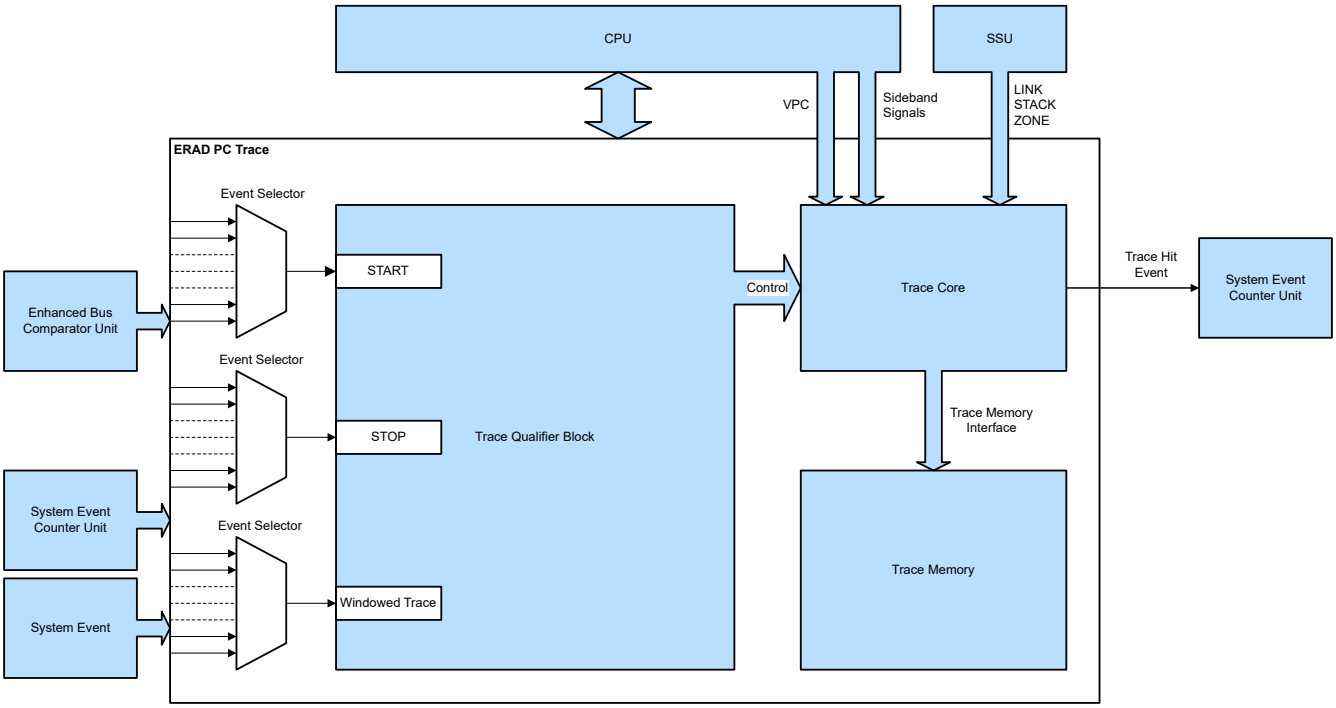


Figure 17-5. PC Trace Block Diagram

17.4.2 Trace Qualification Modes

There are three modes of trace qualification available:

1. Normal mode: In this mode, trace is enabled without any qualifiers. Trace control happens purely through software writing to the PCTRACE_GLOBAL register. When operating PC Trace in normal mode, The PCTRACE_LOGPC_SOFTENABLE captures the program counter value at the point when PCTRACE_GLOBAL.EN is set to 1, and the PCTRACE_LOGPC_SOFTDISABLE register captures the program counter value at the point PCTRACE_GLOBAL.EN is set to 0.
2. Windowed Trace mode: In windowed mode, PC Trace can be activated or stopped based on the value of a signal coming from the Event Bus Comparator, System Event Counter, or other system events. The WINDOWED_INP_SEL field in the PCTRACE_QUAL1 register specifies the input signal used to qualify PC Trace operation in windowed mode. By default trace is active when the input signal is high and inactive when the input signal is low; this behavior can be reversed by setting the PCTRACE_QUAL1.WINDOWED_INP_INV bit high.
3. Start-Stop mode: In this mode, one input signal starts the PC trace operation, and a different input signal stops the trace operation. Event Bus Comparator signals, System Event Counter signals and other system event signals can be used as inputs to start or stop the PC Trace. The START_INP_SEL and STOP_INP_SEL fields in the PCTRACE_QUAL2 register specify the input signals used to qualify PC Trace operation in start-stop mode. Once a start event arrives and PC trace operation begins, the PC Trace module ignores all further start events until a stop event is received. Trace start and stop operations are triggered on the rising edge of the input event; this behavior can be reversed by setting the START_INP_INV and STOP_INP_INV bits in the PCTRACE_QUAL1 register.

To set the PC Trace operation mode, write to the TRACE_MODE bit in the PCTRACE_QUAL1 register.

17.4.2.1 Trace Input Signal Conditioning

The PC Trace module provides input conditioning options for signals that are used to qualify operation in Start-Stop or Windowed modes. For each of these (START, STOP, WINDOWED), there is a two-stage synchronizer for asynchronous input signals, and an inverter. The PC Trace input conditioning options are controllable using the PCTRACE_QUAL1 register with the following bits:

- WINDOWED_INP_INV: This bit inverts the input signal for Windowed trace mode selected in WINDOWED_INP_SEL. When set to 1, the trace operation starts on the falling edge of the input, and stops the rising edge of the input. When this bit is set to 0, the trace operation starts on the rising edge of the input, and stops on the falling edge of the input.
- START_INP_INV: This bit inverts the input signal for trace START operation selected in PCTRACE_QUAL2.START_INP_SEL. When set to 1, the trace operation starts on the falling edge of the input. When this bit is set to 0, the trace operation starts on the rising edge of the input.
- STOP_INP_INV: This bit inverts the input signal for trace STOP operation selected in PCTRACE_QUAL2.STOP_INP_SEL. When set to 1, the trace operation stops on the falling edge of the input. When this bit is set to 0, the trace operation stops on the rising edge of the input.

Figure 17-6 describes the signal conditioning circuit available for each input qualifier signal.

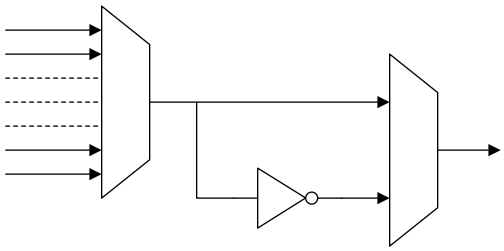


Figure 17-6. Trace Qualifier Input Conditioning Circuit

17.4.3 Trace Memory

PC Trace memory is a 32-bit-wide read-only memory buffer that holds each 31-bit PC value, together with a security BLOCKED status bit. The memory-map section of the device data sheet specifies the size of the PC Trace memory buffer. Trace memory entries are stored in pairs: the discontinuity source and destination addresses. Table 17-2 describes the bit field structure of each trace memory entry.

Table 17-2. Trace Memory Entry Bit Fields

Bits	Field Name	Description
31:1	PROGRAM_COUNTER	Program counter source or destination address value where discontinuity occurred
0	BLOCKED	1 = PROGRAM_COUNTER[31:1] is invalid due to security permissions 0 = PROGRAM_COUNTER[31:1] is valid

Note

For addresses that are blocked due to security zone restrictions, the PROGRAM_COUNTER value is set to 0.

Trace memory is intended solely for debug purposes and does not have parity or Error Correction Code (ECC) support.

17.4.4 PC Trace Software Operation

An example software sequence to perform PC Trace operation is as follows:

1. Initialize the PC Trace module by writing 1 to PCTRACE_GLOBAL.INIT. The trace initialize operation resets the buffer pointer and overflow flags, and clears the values of the PCTRACE_LOGPC_SOFTENABLE and PCTRACE_LOGPC_SOFTDISABLE registers.
2. Start the PC Trace operation by writing 1 to PCTRACE|_GLOBAL.EN.
3. Execute the desired code sequence to be profiled.
4. Stop the PC Trace operation by writing 0 to PCTRACE_GLOBAL.EN. This step is optional, as the PC Trace buffer can be read while trace operation is active.
5. To determine if there are valid entries in the trace buffer and how many:
   a. Examine the buffer pointer by reading PCTRACE_BUFFER.PTR. If the pointer value is 0, then no discontinuities have been detected since PC Trace was initialized. A non-zero value indicates the number of locations in the buffer that contain valid entries. For instance, PTR = 4 indicates that locations 0, 1, 2, and 3 contain valid entries (SRC, DST, SRC, DST).
   b. Examine the BUFFER_FULL bit in the PCTRACE_BUFFER register. If BUFFER_FULL = 1, then the following possibilities apply:
      - If PTR = 0, then the buffer is simply full and contains the maximum number of entries possible.
      - If PTR > 0, then a buffer overflow has occurred. The value of PTR indicates by how many entries the buffer has overflowed: this is a circular buffer.
6. Read PCTRACE_LOGPC_SOFTENABLE (and optionally PCTRACE_LOGPC_SOFTDISABLE) if needed to determine the bounding addresses of the trace operation, in case the code sequence being traced does not begin or end at a discontinuity boundary (for example, partial function trace).
7. Trace discontinuities are recorded in the trace buffer in pairs (SRC, DST). Use this data to reconstruct the code execution sequence. While reading the trace buffer, be sure to examine the BLOCKED status bit to confirm the validity of each trace entry.

17.4.5 Trace Operation in Debug Mode

PC Trace is designed to capture data while the CPU is executing code. Debugger operations such as halt, step, and manual PC modification can compromise the reliability of PC trace data collection. Only use or interpret PC Trace data in the context of a continuous CPU run without debugger interruption or intervention.

CPU execution does not preclude debugger operation of the PC trace module if the debugger owns the PCTRACE module. While the CPU is running and executing code, the debugger can read and write PC Trace registers, read the trace buffer, and enable or disable PC trace operation. Debugger can update the PC Trace registers if debugger owns PCTRACE module. The PCTRACE_LOGPC_SOFTENABLE and PCTRACE_LOGPC_SOFTDISABLE registers record the start and stop PC addresses to provide accurate trace window information during manual trace starts or stops triggered by writing to the PCTRACE_GLOBAL.EN bit.

17.5 ERAD Ownership, Initialization, and Reset

Although the features of the ERAD module are typically used by the debugger, user applications can also take advantage of the capabilities to monitor buses and generate interrupts and events. ERAD's features such as the EBC, SEC, ANDOR, and PCTRACE have a dedicated ownership scheme and register set for every instance. There are three possible ownership scenarios:

1. The user elects to completely hand over the ownership of the ERAD features to the application software or the debugger.
2. Only the current owner of the module (application code or debugger) is allowed to use the module at a given time. Either the debugger or application own a particular EBC/SEC block to ensure no conflicts arising.
3. There is no ERAD features. In this mode, both application code and debugger can access the module at any given time. It is critical for the software, both on the application side and the debugger side, to resolve any potential conflicts. An example scenario in this mode can be for the debugger to use some of the EBC and SEC units, while the application software uses the remaining units.

The ERAD module initializes the internal states and all registers to the initial/reset states under the following conditions:

- At power-on-reset (POR)
- With DCON and SYSRSN
  - Debug logic disconnected when the debugger owns the module
  - Functional reset when application owns the module

17.5.1 Feature Level Ownership

For each feature such as the EBC, SEC, Masks, and PCTrace an ownership scheme is implemented by specifying if the debugger or application owns the feature. By default the Owner register belongs to no owner and ZONE register is set to ZONE0.

Scenario 1 Basic use: Debugger Ownership

1. Write to OWNER register of 0x01 is permitted since existing value was 0x00
   - Writes to OWNER register is successful and ZONE register remains at 0
2. Read back and check if OWNER register is 0x01
3. Debugger can configure and start using the block
   - If application accidentally writes to OWNER register, accesses are blocked during this time
4. Debugger completes work and a write to the OWNER register of 0x00 releases this block
   - Writes are successful and OWNER register is 0x00
   - ZONE register is 0x0000

Scenario 2: Application Ownership

1. Write to OWNER register of 0x2 is permitted since existing value was 0x00
   - Writes to OWNER register is successful and ZONE register is loaded with ZONE ID from CPU
   - Write to this register emerged from ZONE2, so ZONE register gets loaded with 0x2
2. Read back and check if OWNER register is 0x2
3. Application code from ZONE2 can configure and start using the block
   - If application code from another ZONE tries to write to OWNER register, the write is blocked
   - If debugger tries to write to OWNER register, the write is blocked
   - Application code from ZONE2 completes work and writes OWNER register to 0x00
4. Writes are successful and OWNER register becomes 0x00

These two scenarios are the flow of the feature level ownership. For application code to identify whether the application has ownership a simple read to the Semaphore register can indicate the status. Debugger has no notion of ZONES, only CPU does.

17.5.2 Feature Access Security Mechanism

Ownership	Registers Writable?	Registers Readable?
No Owner	No	Ownership register alone is always readable by debugger and the application
Debugger	Writes are allowed by debugger. Writes are blocked by application.	Reads are allowed by debugger. Application reads are blocked, returns all zero data.
Application	Writes are blocked by debugger. Writes are allowed by application if access and owner come from same ZONE. In addition if CPU1.LINK2 owns the block and access originated from CPU1.LINK2. Writes are blocked if access and owner from different ZONE or if SROOT owns block but access originated from non CPU1.LINK2 location.	Reads are allowed by debugger only if corresponding owner ZONE is enabled for debug. Application reads are allowed if access and owner come from same ZONE. In addition if CPU1.LINK2 owns block and access originated from CPU1.LINK2. Reads are blocked by application if access and owner come from different ZONEs or if CPU1.LINK2 owns block but access originated from non CPU1.LINK2 location.

Note

ERAD security is based on the granularity of ZONEs only (Not LINKs). Only exception is the CPU1.LINK2 (security root LINK). If an ERAD feature is owned by CPU1.LINK2, then there is no security check performed and all the valid accesses generate triggers irrespective of which zone performed that access (Any matching access from ZONE0, ZONE1, ZONE2 or ZONE3 can generate a trigger if that particular feature is owned by CPU1.LINK2).

Table 17-3. EBC Specific Function and Security Mechanism

Ownership	All EBC Event Except Hardware Breakpoints(1)
Application (CPU1.LINK2 not configured)	• EBC events are generated when the EBC is owned by a given ZONE and CPU is executing code from the same ZONE • EBC events are NOT generated when the EBC is owned by a given ZONE and CPU is executing code from a different ZONE, irrespective of that ZONE being enabled for debug or not.
Application (CPU1.LINK2)	• EBC events are ALWAYS generated when the EBC is owned by CPU1.LINK2 • CPU code execution ZONE is ignored • ZONE debug enables are also ignored
Debug	• EBC events are generated when CPU is executing code from a ZONE that is enabled for debug • If an environment to debug everything like the CPU1.LINK2 equivalent in Debug mode is needed, enable all the zones for debug

(1) Hardware breakpoints tags are generated always irrespective of ownership/security. CPU takes care of handling the breakpoints accordingly.

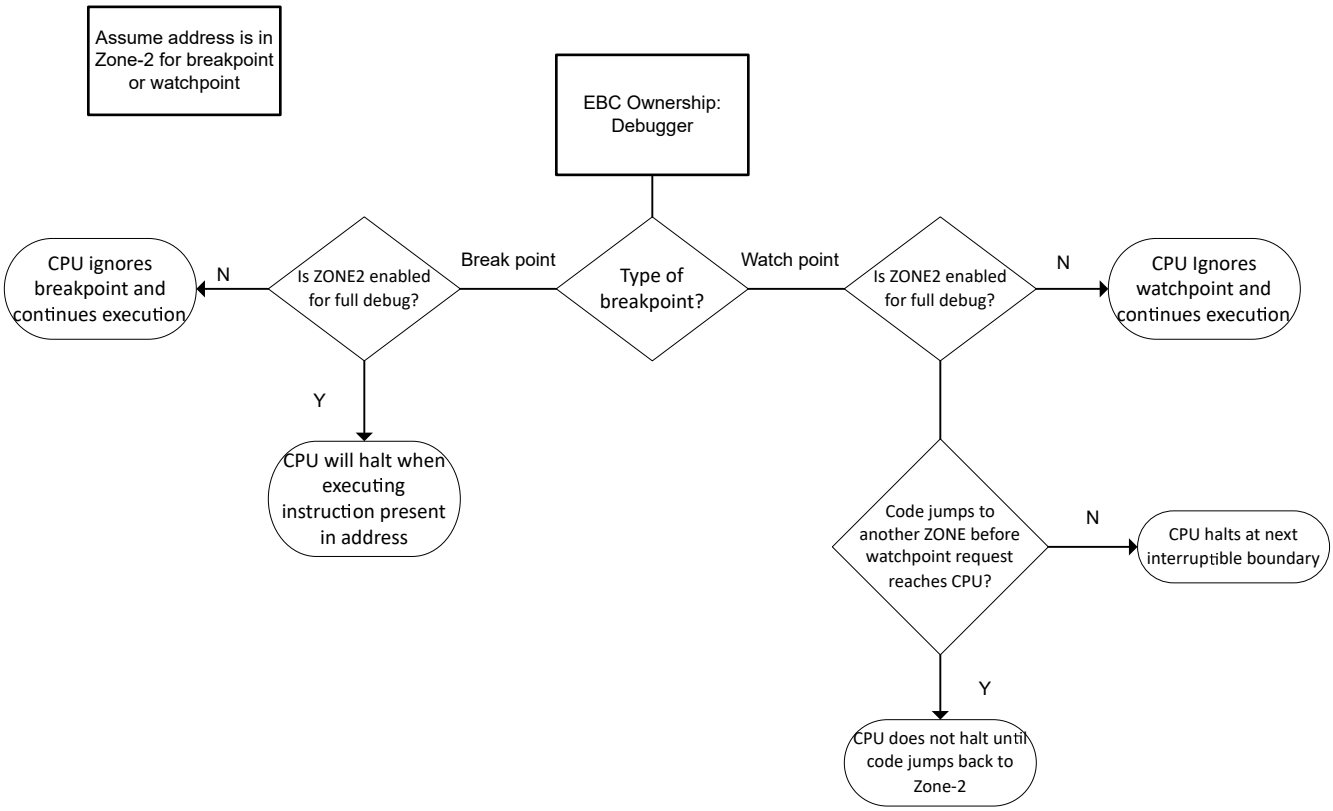


Figure 17-7. EBC Owned by Debugger

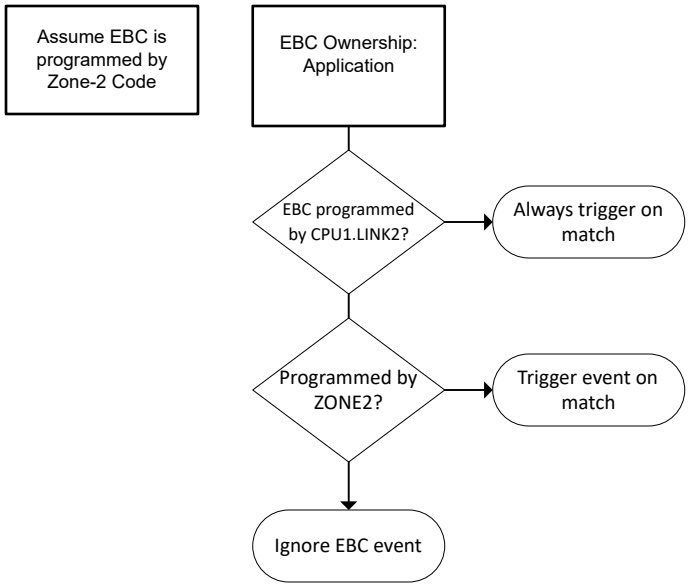


Figure 17-8. EBC Owned by Application

17.5.3 PC Trace Access Security Mechanism

Irrespective of which zone is executing code, trace values are always dumped when there is discontinuity. Only when memory is read back security restrictions apply. In a nutshell if application code is reading the trace memory, only the discontinuities stored for the zone from where code is executing is sent out and other are blocked. If the ZONE for which the discontinuity is stored is enabled for debug, then the trace memory is readable.

For example, if reads are initiated to the trace RAM by code originating from SROOT LINK, then all the entries present in the trace RAM are accessible, irrespective of which zone that discontinuity belonged to. Otherwise based on security permissions, either the discontinuity address is read if allowed or is zeroed out with the blocked status bit set.

17.6 ERAD Programming Sequence

The ERAD module can be used to set hardware breakpoints and hardware watch points. The programming sequences to set hardware breakpoints, hardware watch points, or to use the timers to profile and analyze the system are described in this section. The same sequences can be used by both the debugger software and user application code.

Refer to the Driverlib example projects in C2000Ware for JavaScript files to configure the ERAD module. Example projects are also available to showcase the usage of these script files. These examples can be found in the driverlib\\examples\erad directory under the C2000Ware installation directory.

17.6.1 Hardware Breakpoint and Hardware Watch Point Programming Sequence

The programming sequence is identical when using the EBC units, regardless of whether the debug software or the application is programing the units. A typical programming sequence for a unit is:

- Read and make sure the ownership is set as expected; if not, acquire the ownership before proceeding further if required.
- Make sure the unit is in IDLE mode.
- Set up the address reference, mask, bus select and stop bits.
- Enable the corresponding module bit in the global enable register.
- Once the usage is completed, write 1 to clear the EVENT_FIRED sticky bit. This takes the module back to the enabled state.

The example programming sequences for hardware breakpoints and hardware watch points are:

Set a hardware breakpoint on address 0x201000:

- Read OWNER to confirm ownership.
- Read EBC_STATUS to confirm the module is in IDLE state.
- Write 0x0 to EBC_MASKH/MASKL.
- Write 0x201000 to EBC_REFH/REFL.
- Enable the corresponding module bit in the EBC enable register.

Set a hardware watch point on read of addresses from 0x121010 to 0x12101F:

- Read OWNER to confirm ownership.
- Read EBC_STATUS to confirm the module is in IDLE state.
- Write 0xF to EBC_MASKH/MASKL.
- Write 0x121010 to EBC_REFH/REFL.
- Enable the corresponding module bit in the global enable register.

Set a hardware watch point on write to address 0xFF10101A:

- Read OWNER to confirm ownership
- Read EBC_STATUS to confirm the module is in IDLE state
- Write 0x0 to EBC_MASKH/MASKL
- Write 0xFF10101A to EBC_REFH/REFL
- Enable the corresponding module bit in the global enable register.

17.6.2 Timer and Counter Programming Sequence

The programming sequence is identical when using the SEC units, regardless of whether the debug software or the application is programing the units. Typical programming sequence for a unit is:

- Read and make sure the ownership is set as expected. If not, acquire the ownership before proceeding further if required.
- Make sure the unit is in IDLE mode.
- Set up the counter reference, counter registers (clear/reset if a clean start is required).
- Enable the corresponding module bit in the enable register.
- Once the usage is completed, write 1 to clear the EVENT_FIRED sticky bit. This takes the module back to the enabled state.

Set up a free running counter:

- Read and make sure the ownership is set as expected. If not, acquire the ownership before proceeding further, if required.
- Read SEC_STATUS to confirm that the module is in IDLE state.
- Write 0x0 to CNT_INP_SEL_EN.
- Write 0x0 to SEC_CNTL.
- Enable the module in the SEC_CNTL.EN register.

Set up the counter to count the duration spent between addresses 0x1000 and 0x1210:

- Read and make sure the ownership is set as expected. If not, acquire the ownership before proceeding further, if required.
- Read SEC_STATUS to confirm that the module is in IDLE state.
- Set up the EBC unit 1 to generate an event for VPC = 0x1000.
- Set up the EBC unit 2 to generate an event for VPC = 0x1210.
- Enable the module in the SEC_CNTL.EN register.

Set up the counter to count the number of times a function at address 0x2010 is called and fire an interrupt if this count reaches 0x300:

- Read and make sure the ownership is set as expected. If not, acquire the ownership before proceeding further, if required.
- Read SEC_STATUS to confirm that the module is in IDLE state.
- Set up the EBC unit 1 to generate an event for VPC = 0x2010.
- Write 0x300 SEC_REF.
- Enable the counter in the EDGE_LEVEL, and also allow the counter to generate an interrupt when the count matches the reference. This is achieved by writing 0x42 to SEC_CNTL.
- Enable the module in the SEC_CNTL.EN register.

17.7 Software

17.7.1 ERAD Registers to Driverlib Functions

Table 17-4. ERAD Registers to Driverlib Functions

File	Driverlib Function
GLBL_ERAD_ID	
-	
GLBL_EVENT_STAT	
erad.h	ERAD_getEventStatus
EBC_OWNER(i)	
erad.h	ERAD_setOwnership
erad.h	ERAD_getBusCompOwnership
erad.h	ERAD_setBusCompOwnership
EBC_CNTL(i)	
erad.c	ERAD_configBusComp
erad.h	ERAD_enableModules
erad.h	ERAD_disableModules
erad.h	ERAD_enableBusCompModule
erad.h	ERAD_disableBusCompModule
erad.h	ERAD_enableNMI
erad.h	ERAD_disableNMI
EBC_STATUS(i)	
erad.h	ERAD_getBusCompStatus
erad.h	ERAD_clearBusCompEvent
EBC_STATUSCLEAR(i)	
erad.h	ERAD_clearBusCompEvent
EBC_REFL(i)	
erad.c	ERAD_configBusComp
EBC_REFH(i)	
erad.c	ERAD_configBusComp
EBC_MASKL(i)	
erad.c	ERAD_configBusComp
EBC_MASKH(i)	
erad.c	ERAD_configBusComp
EBC_WP_PC(i)	
erad.h	ERAD_getWatchpointPC
SEC_OWNER(i)	
erad.h	ERAD_setOwnership
erad.h	ERAD_getCounterOwnership
erad.h	ERAD_setCounterOwnership
SEC_CNTL(i)	
erad.c	ERAD_configCounterInCountingMode
erad.c	ERAD_configCounterInStartStopMode
erad.c	ERAD_configCounterInCumulativeMode
erad.h	ERAD_enableModules
erad.h	ERAD_disableModules
erad.h	ERAD_enableCounterModule

Table 17-4. ERAD Registers to Driverlib Functions (continued)

File	Driverlib Function
erad.h	ERAD_disableCounterModule
erad.h	ERAD_enableNMI
erad.h	ERAD_disableNMI
erad.h	ERAD_enableCounterResetInput
erad.h	ERAD_disableCounterResetInput
SEC_STATUS(i)	
erad.h	ERAD_getCounterStatus
erad.h	ERAD_clearCounterEvent
erad.h	ERAD_clearCounterOverflow
SEC_STATUSCLEAR(i)	
erad.h	ERAD_clearCounterEvent
erad.h	ERAD_clearCounterOverflow
SEC_REF(i)	
erad.c	ERAD_configCounterInCountingMode
erad.c	ERAD_configCounterInStartStopMode
erad.c	ERAD_configCounterInCumulativeMode
SEC_INPUT_SEL1(i)	
erad.c	ERAD_configCounterInCountingMode
erad.c	ERAD_configCounterInStartStopMode
erad.c	ERAD_configCounterInCumulativeMode
erad.h	ERAD_enableCounterResetInput
SEC_INPUT_SEL2(i)	
erad.c	ERAD_configCounterInStartStopMode
erad.c	ERAD_configCounterInCumulativeMode
SEC_INPUT_COND(i)	
erad.h	ERAD_setCounterInputConditioning
SEC_COUNT(i)	
erad.h	ERAD_getCurrentCount
erad.h	ERAD_setCurrentCount
SEC_MAX_COUNT(i)	
erad.h	ERAD_getMaxCount
erad.h	ERAD_setMaxCount
SEC_MIN_COUNT(i)	
erad.h	ERAD_getMinCount
erad.h	ERAD_setMinCount
AND_MASK_OWNER(i)	
erad.h	ERAD_setOwnership
erad.h	ERAD_getMaskOwnership
erad.h	ERAD_setMaskOwnership
AND_MASK_CTL(i)	
erad.c	ERAD_configMask
EVENT_AND_MASK(i)	
erad.c	ERAD_configMask
OR_MASK_OWNER(i)	
erad.h	ERAD_setOwnership

Table 17-4. ERAD Registers to Driverlib Functions (continued)

File	Driverlib Function
erad.h	ERAD_getMaskOwnership
erad.h	ERAD_setMaskOwnership
OR_MASK_CTL(i)	
erad.c	ERAD_configMask
EVENT_OR_MASK(i)	
erad.c	ERAD_configMask
PCTRACE_OWNER	
erad.h	ERAD_setOwnership
erad.h	ERAD_getPCTraceOwnership
erad.h	ERAD_setPCTraceOwnership
PCTRACE_GLOBAL	
erad.h	ERAD_enablePCTrace
erad.h	ERAD_disablePCTrace
erad.h	ERAD_initPCTraceBuffer
PCTRACE_BUFFER	
-	
PCTRACE_QUAL1	
erad.h	ERAD_setPCTraceMode_NoQualifiers
erad.h	ERAD_setPCTraceMode_Windowed
erad.h	ERAD_setPCTraceMode_StartStop
PCTRACE_QUAL2	
erad.h	ERAD_setPCTraceMode_NoQualifiers
erad.h	ERAD_setPCTraceMode_Windowed
erad.h	ERAD_setPCTraceMode_StartStop
PCTRACE_LOGPC_SOFTENABLE	
-	
PCTRACE_LOGPC_SOFTDISABLE	
-	
PCTRACE_BUFFER_BASE(i)	
-	

17.8 ERAD Registers

This section describes the Embedded Real-Time Analysis and Diagnostic Registers.

17.8.1 ERAD Base Address Table

Table 17-5. ERAD Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
ERAD_REGS	ERAD_BASE	0x3003_0000	-	YES	YES	YES	-	-	-	YES

17.8.1.1 ERAD_REGS Registers

Table 17-6 lists the memory-mapped registers for the ERAD_REGS registers. All register offset addresses not listed in Table 17-6 should be considered as reserved locations and the register contents should not be modified.

Table 17-6. ERAD_REGS Registers

Offset	Acronym	Register Name	Protection
0h	GLBL_ERAD_ID	Debug Peripheral ID	
4h	GLBL_EVENT_STAT	Global Event Status Register	
40h + formula	EBC_OWNER_j	EBC Owner Register	
44h + formula	EBC_CNTL_j	EBC Control Register	
48h + formula	EBC_STATUS_j	EBC Status Register	
4Ch + formula	EBC_STATUSCLEAR_j	EBC Clear Register	
50h + formula	EBC_REFL_j	EBC Reference Low Register	
54h + formula	EBC_REFH_j	EBC Reference High Register	
58h + formula	EBC_MASKL_j	EBC Mask Low Register	
5Ch + formula	EBC_MASKH_j	EBC Mask High Register	
60h + formula	EBC_WP_PC_j	EBC Watchpoint PC Register	
440h + formula	SEC_OWNER_j	SEC Owner Register	
444h + formula	SEC_CNTL_j	SEC Control Register	
448h + formula	SEC_STATUS_j	SEC Status Register	
44Ch + formula	SEC_STATUSCLEAR_j	SEC Clear Register	
450h + formula	SEC_REF_j	SEC Reference Register	
454h + formula	SEC_INPUT_SEL1_j	SEC Input Select Register1	
458h + formula	SEC_INPUT_SEL2_j	SEC Input Select Register2	
45Ch + formula	SEC_INPUT_COND_j	SEC Input Conditioning Register	
460h + formula	SEC_COUNT_j	SEC Counter Register	
464h + formula	SEC_MAX_COUNT_j	SEC Max Count Register	
468h + formula	SEC_MIN_COUNT_j	SEC Min Count Register	
640h + formula	AND_MASK_OWNER_j	AND Owner Register	
644h + formula	AND_MASK_CTL_j	AND Control Register	
648h + formula	EVENT_AND_MASK_j	AND Event Selection Register	

Table 17-6. ERAD_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
740h + formula	OR_MASK_OWNER_j	OR Owner Register	
744h + formula	OR_MASK_CTL_j	OR Control Register	
748h + formula	EVENT_OR_MASK_j	OR Event Selection Register	
840h	PCTRACE_OWNER	Owner Register	
844h	PCTRACE_GLOBAL	Global Control Register	
848h	PCTRACE_BUFFER	Trace Buffer pointer register	
84Ch	PCTRACE_QUAL1	Trace Qualifier register 1	
850h	PCTRACE_QUAL2	Trace Qualifier register 2	
854h	PCTRACE_LOGPC_SOFTENABLE	PC when PC Trace was last enabled by software	
858h	PCTRACE_LOGPC_SOFTDISABLE	PC when PC Trace was last disabled by software	
1000h + formula	PCTRACE_BUFFER_BASE_y	Trace Buffer Base address	

Complex bit access types are encoded to fit into small table cells. Table 17-7 shows the codes that are used for access types in this section.

Table 17-7. ERAD_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

17.8.1.1.1 GLBL_ERAD_ID Register (Offset = 0h) [Reset = 62090000h]

GLBL_ERAD_ID is shown in Figure 17-9 and described in Table 17-8.

Return to the Summary Table.

Debug Peripheral ID

Figure 17-9. GLBL_ERAD_ID Register

31	30	29	28	27	26	25	24
RESERVED		RESERVED		FUNC			
R-1h		R-2h		R-209h			
23	22	21	20	19	18	17	16
FUNC							
R-209h							
15	14	13	12	11	10	9	8
RESERVED						MAJOR	
R-0h						R-0h	
7	6	5	4	3	2	1	0
CUSTOM		MINOR					
R-0h		R-0h					

Table 17-8. GLBL_ERAD_ID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	1h	Reserved
29-28	RESERVED	R	2h	Reserved
27-16	FUNC	R	209h	0x209 => C29 Embedded Real-Time Analysis and Diagnostics Reset type: SYSRSn
15-11	RESERVED	R	0h	Reserved
10-8	MAJOR	R	0h	Based on RTL Version Reset type: SYSRSn
7-6	CUSTOM	R	0h	Based on RTL Version Reset type: SYSRSn
5-0	MINOR	R	0h	Based on RTL Version Reset type: SYSRSn

17.8.1.1.2 GLBL_EVENT_STAT Register (Offset = 4h) [Reset = 00000000h]

GLBL_EVENT_STAT is shown in Figure 17-10 and described in Table 17-9.

Return to the Summary Table.

Global Event Status Register

Figure 17-10. GLBL_EVENT_STAT Register

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

Table 17-9. GLBL_EVENT_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SEC	R	0h	This bit directly reflects the state of the EVENT_FIRED bit of the SEC unit x. 0 No Event 1 Event Fired Bit-16 SEC1 Bit-17 SEC2 .... Bit-(n-15) SECn Reset type: ERAD_RESET
15-0	EBC	R	0h	This bit directly reflects the state of the EVENT_FIRED bit of the EBC unit x. 0 No Event 1 Event Fired Bit-0 EBC1 Bit-1 EBC2 .... Bit-n EBCn Reset type: ERAD_RESET

17.8.1.1.3 EBC_OWNER_j Register (Offset = 40h + formula) [Reset = 00000000h]

EBC_OWNER_j is shown in Figure 17-11 and described in Table 17-10.

Return to the Summary Table.

EBC Owner Register

Offset = 40h + (j * 40h); where j = 0h to 7h

Figure 17-11. EBC_OWNER_j Register

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

Table 17-10. EBC_OWNER_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18-17	CONFIG	R	0h	00 => EBC Full 01 => EBC Lite Others => Reserved Reset type: ERAD_RESET
16	SEM	R	0h	1 => Access and owner from same zone (Additionally if SROOT LINK owns the block and access originated from SROOTLINK area) 0 => Access and owner from different zone (Additionally if SROOT LINK owns the block, but access originated from a non SROOT LINK area) These bits are only valid when OWNER = 10 (App owned) Reset type: ERAD_RESET
15-13	RESERVED	R	0h	Reserved
12	SROOT	R	0h	0 => This block is not owned by SROOT LINK 1 => This block is owned by SROOT LINK Reset type: ERAD_RESET
11-8	ZONE	R	0h	0000 => Zone-0 0001 => Zone-1 0010 => Zone-2 0011 => Zone-3 All others => Reserved These bits are only valid when OWNER = 10 (App owned) Reset type: ERAD_RESET
7-2	RESERVED	R	0h	Reserved

Table 17-10. EBC_OWNER_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	OWNER	R/W	0h	00 => No Owner 01 => Debug Owned 10 => App Owned 11 => Reserved Debugger : Read-> Returns current ownership ID Writes -> Allowed only if current value is 00 or 01 Application: Read-> Returns current ownership ID Writes -> Allowed only if current value is 00 or 10 NOTE : When a 00 gets successfully written, all the rest of control/ status registers gets reset/cleared Reset type: ERAD_RESET

17.8.1.1.4 EBC_CNTL_j Register (Offset = 44h + formula) [Reset = 00000000h]

EBC_CNTL_j is shown in Figure 17-12 and described in Table 17-11.

Return to the Summary Table.

EBC Control Register

Offset = 44h + (j * 40h); where j = 0h to 7h

Figure 17-12. EBC_CNTL_j Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						SPSEL	
R-0h						R/W-0h	
15	14	13	12	11	10	9	8
SPSEL		SPSEL_MATCH_EN	STACK_QUAL	COMP_MODE			NMI_EN
R/W-0h		R/W-0h	R/W-0h	R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
INTERRUPT	HALT	BUS_SEL					EN
R/W-0h	R/W-0h	R/W-0h					R/W-0h

Table 17-11. EBC_CNTL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17-14	SPSEL	R/W	0h	Stack pointer Select value to be used for SPSEL match. Reset type: ERAD_RESET
13	SPSEL_MATCH_EN	R/W	0h	0 = SPSEL match disabled 1 = SPSEL match enabled This bit is relevant only for data read and data write address comparison. When enabled, SPSEL match will additionally be considered to generate a match. Reset type: ERAD_RESET
12	STACK_QUAL	R/W	0h	0 = Stack access qualifier disabled 1 = Stack access qualifier enabled This bit is relevant only for data read and data write address comparison. When enabled, the corresponding stack access qualifier will additionally be considered to generate a match. Reset type: ERAD_RESET
11-9	COMP_MODE	R/W	0h	EBC compare modes: 000 Regular masked compare EBC_MSK will be ignored for the following modes: 100 Bus value GT EBC_REF 101 Bus value GE EBC_REF 110 Bus value LT EBC_REF 111 Bus value LE EBC_REF GT means Greater Than GE means Greater or Equal LT means Less Than LE means Lesser or Equal Reset type: ERAD_RESET

Table 17-11. EBC_CNTL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	NMI_EN	R/W	0h	0 = Will assert ERAD interrupt 1 = Will assert NMI (With only exception of BUS_SEL = PAB, which will not generate any interrupt or event or NMI. This is only used for breakpoints by debug controller) Reset type: ERAD_RESET
7	INTERRUPT	R/W	0h	This bit decides whether the EBC unit will generate interrupt when event matches occur. Note that the event outputs will always be generated regardless of the state of this bit. 0 The EBC unit will not cause any action towards the CPU. 1 The EBC unit will assert NMI/ERAD interrupt based on NMI_EN bit for matching data accesses (With only exception of BUS_SEL = PAB, which will not generate any interrupt or event or NMI. This is only used for breakpoints by debug controller) Reset type: ERAD_RESET
6	HALT	R/W	0h	This bit decides whether the EBC unit will generate CPU halting signals when event matches occur. Note that the event outputs will always be generated regardless of the state of this bit. 0 The EBC unit will not cause any action towards halting the CPU. 1 The EBC unit will assert break tags for matching program fetches(Breakpoint) and Halt request rest of the matching accesses(Watchpoint). These can cause the CPU to HALT Reset type: ERAD_RESET
5-1	BUS_SEL	R/W	0h	00000 PAB for instruction fetches 00001 DWAB for data write accesses 00101 DRAB_W address aligned with RDATA1/RDATA2 (either DRAB1_W/DRAB2_W) 00110 DRAB1_W address aligned with RDATA1 00111 DRAB2_W address aligned with RDATA2 01000 DWDB for write data match 01001 DRDB for read data match (either DRDB1/2) 01010 DRDB1 for read data match (RDATA1) 01011 DRDB2 for read data match (RDATA1) 01100 VPC Instruction aligned match (VPC for R1 phase) 01110 VPC W aligned match (VPC for Write address/data phase/ Read data phase) Others Reserved Reset type: ERAD_RESET
0	EN	R/W	0h	0 = EBC Disabled 1 = EBC Enabled Reset type: ERAD_RESET

17.8.1.1.5 EBC_STATUS_j Register (Offset = 48h + formula) [Reset = 00000000h]

EBC_STATUS_j is shown in Figure 17-13 and described in Table 17-12.

Return to the Summary Table.

EBC Status Register

Offset = 48h + (j * 40h); where j = 0h to 7h

Figure 17-13. EBC_STATUS_j Register

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

Table 17-12. EBC_STATUS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	Reserved
9-8	STATUS	R	0h	EBC status: 00 Idle 10 Enabled 11 Completed Reset type: ERAD_RESET
7-1	RESERVED	R	0h	Reserved
0	EVENT_FIRED	R	0h	This is a sticky bit which gets set every time the EBC unit generates a match event. This will be used by software to figure out whether this EBC module fired an event or not. This bit will get cleared by writing a '1' to bit 0 of the EBC_STATUSCLEAR register. Reset type: ERAD_RESET

17.8.1.1.6 EBC_STATUSCLEAR_j Register (Offset = 4Ch + formula) [Reset = 00000000h]

EBC_STATUSCLEAR_j is shown in Figure 17-14 and described in Table 17-13.

Return to the Summary Table.

EBC Clear Register

Offset = 4Ch + (j * 40h); where j = 0h to 7h

Figure 17-14. EBC_STATUSCLEAR_j Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							EVENT_FIRED
R-0h							W1C-0h

Table 17-13. EBC_STATUSCLEAR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	EVENT_FIRED	W1C	0h	Event Clear register: 0 No action. 1 A write with this bit set to 1 will clear the sticky EVENT_FIRED bit in the EBC_STATUS register and bring the EBC statemachine status back to ENABLED state if EBC is enabled or IDLE state if EBC is disabled. Reads of this bit position will always return a 0. Note : If hardware is trying to set EVENT_FIRED bit at the same cycle software is trying to clear, then the clear will be ignored and status will remain set Reset type: ERAD_RESET

17.8.1.1.7 EBC_REFL_j Register (Offset = 50h + formula) [Reset = 00000000h]

EBC_REFL_j is shown in Figure 17-15 and described in Table 17-14.

Return to the Summary Table.

EBC Reference Low Register

Offset = 50h + (j * 40h); where j = 0h to 7h

Figure 17-15. EBC_REFL_j Register

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

Table 17-14. EBC_REFL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REF	R/W	0h	This register contains the lower 32-bits of the 64-bit reference used for comparison. The contents of this register are used along with the mask register to determine the address match. The equation used to determine a match is as follows. Match is true if, (compare bus | mask) == (reference | mask) This register is writable by CPU only if application owns the unit. Otherwise, the writes are ignored. The register is writable by the debugger only if the debugger owns this unit. Otherwise, debugger writes are ignored. Note : compare bus = {REFH[31:0],REFL{31:0} REFH and REFL are used in conjunction for 64-bit bus compares. So always disable, program the reference registers H,L and then re-enable for 64-bit comparison. Reset type: ERAD_RESET

17.8.1.1.8 EBC_REFH_j Register (Offset = 54h + formula) [Reset = 00000000h]

EBC_REFH_j is shown in Figure 17-16 and described in Table 17-15.

Return to the Summary Table.

EBC Reference High Register

Offset = 54h + (j * 40h); where j = 0h to 7h

Figure 17-16. EBC_REFH_j Register

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

Table 17-15. EBC_REFH_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REF	R/W	0h	This register contains the top 32-bits of the 64-bit reference used for comparison. The contents of this register are used along with the mask register to determine the address match. The equation used to determine a match is as follows. Match is true if, (compare bus | mask) == (reference | mask) This register is writable by CPU only if application owns the unit. Otherwise, the writes are ignored. The register is writable by the debugger only if the debugger owns this unit. Otherwise, debugger writes are ignored. Note : compare bus = {REFH[31:0],REFL{31:0} REFH and REFL are used in conjunction for 64-bit bus compares. So always disable, program the reference registers H,L and then re-enable for 64-bit comparison. Reset type: ERAD_RESET

17.8.1.1.9 EBC_MASKL_j Register (Offset = 58h + formula) [Reset = 00000000h]

EBC_MASKL_j is shown in Figure 17-17 and described in Table 17-16.

Return to the Summary Table.

EBC Mask Low Register

Offset = 58h + (j * 40h); where j = 0h to 7h

Figure 17-17. EBC_MASKL_j Register

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

Table 17-16. EBC_MASKL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This register contains the lower 32-bits of the 64-bit mask used for comparison. The contents of this register are used along with the reference register to determine the address match. The equation used to determine a match is as follows. Match is true if, (compare bus | mask) == (reference | mask) This register is writable by CPU only if application owns the unit. Otherwise, the writes are ignored. The register is writable by the debugger only if the debugger owns this unit. Otherwise, the writes are ignored. Note : mask = {MASKH[31:0],MASKL{31:0} MASKH and MASKL are used in conjunction for 64-bit bus compares. So always disable, program the reference registers H,L and then re-enable for 64-bit comparison. Important Note : If PAB is chosen for comparison and masks are enabled, then the following restriction apply. MASKL[3:0] can only take one of the following values 0x0, 0x1, 0x3, 0x7 & 0xF. Rest of the mask bits can be chosen freely based on need. Reset type: ERAD_RESET

17.8.1.1.10 EBC_MASKH_j Register (Offset = 5Ch + formula) [Reset = 00000000h]

EBC_MASKH_j is shown in Figure 17-18 and described in Table 17-17.

Return to the Summary Table.

EBC Mask High Register

Offset = 5Ch + (j * 40h); where j = 0h to 7h

Figure 17-18. EBC_MASKH_j Register

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

Table 17-17. EBC_MASKH_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This register contains the higher 32-bits of the 64-bit mask used for comparison. The contents of this register are used along with the reference register to determine the address match. The equation used to determine a match is as follows. Match is true if, (compare bus | mask) == (reference | mask) This register is writable by CPU only if application owns the unit. Otherwise, the writes are ignored. The register is writable by the debugger only if the debugger owns this unit. Otherwise, the writes are ignored. Note : mask = {MASKH[31:0],MASKL{31:0} MASKH and MASKL are used in conjunction for 64-bit bus compares. So always disable, program the reference registers H,L and then re-enable for 64-bit comparison. Reset type: ERAD_RESET

17.8.1.1.11 EBC_WP_PC_j Register (Offset = 60h + formula) [Reset = 00000000h]

EBC_WP_PC_j is shown in Figure 17-19 and described in Table 17-18.

Return to the Summary Table.

EBC Watchpoint PC Register

Offset = 60h + (j * 40h); where j = 0h to 7h

Figure 17-19. EBC_WP_PC_j Register

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

Table 17-18. EBC_WP_PC_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PC	R	0h	This register captures the PC of instruction which caused the watchpoint. This is valid for Data read/write related bus selects only Reset type: ERAD_RESET

17.8.1.1.12 SEC_OWNER_j Register (Offset = 440h + formula) [Reset = 00000000h]

SEC_OWNER_j is shown in Figure 17-20 and described in Table 17-19.

Return to the Summary Table.

SEC Owner Register

Offset = 440h + (j * 40h); where j = 0h to 3h

Figure 17-20. SEC_OWNER_j Register

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

Table 17-19. SEC_OWNER_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18-17	CONFIG	R	0h	00 => SEC Full 01 => SEC Lite Others => Reserved Reset type: ERAD_RESET
16	SEM	R	0h	1 => Access and owner from same zone (Additionally if SROOT LINK owns the block and access originated from SROOTLINK area) 0 => Access and owner from different zone (Additionally if SROOT LINK owns the block, but access originated from a non SROOT LINK area) These bits are only valid when OWNER = 10 (App owned) Reset type: ERAD_RESET
15-13	RESERVED	R	0h	Reserved
12	SROOT	R	0h	0 => This block is not owned by SROOT LINK 1 => This block is owned by SROOT LINK Reset type: ERAD_RESET
11-8	ZONE	R	0h	0000 => Zone-0 0001 => Zone-1 0010 => Zone-2 0011 => Zone-3 All others => Reserved These bits are only valid when OWNER = 10 (App owned) Reset type: ERAD_RESET
7-2	RESERVED	R	0h	Reserved

Table 17-19. SEC_OWNER_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	OWNER	R/W	0h	00 => No Owner 01 => Debug Owned 10 => App Owned 11 => Reserved Debugger : Read-> Returns current ownership ID Writes -> Allowed only if current value is 00 or 01 Application: Read-> Returns current ownership ID Writes -> Allowed only if current value is 00 or 10 NOTE : When a 00 gets successfully written, all the rest of control/ status registers gets reset/cleared Reset type: ERAD_RESET

17.8.1.1.13 SEC_CNTL_j Register (Offset = 444h + formula) [Reset = 00000000h]

SEC_CNTL_j is shown in Figure 17-21 and described in Table 17-20.

Return to the Summary Table.

SEC Control Register

Offset = 444h + (j * 40h); where j = 0h to 3h

Figure 17-21. SEC_CNTL_j Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED					FREE_RUN	RST_INP_SEL_EN	CNT_INP_SEL_EN
R-0h					R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
NMI_EN	INTERRUPT	HALT	RST_ON_MATCH	START_STOP_CUMULATIVE	START_STOP_MODE	EDGE_LEVEL	EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 17-20. SEC_CNTL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	Reserved
10	FREE_RUN	R/W	0h	0 = CPU halt is ignored by the SEC counter all times 1 = CPU halt with stop the counter until CPU is halted. i.e. If counter gets enabled by SEC to count and CPU is halted then the counter stops counting until CPU is in halt state. Once the CPU resumes execution out of halt mode, then Counter will be active again. Reset type: ERAD_RESET
9	RST_INP_SEL_EN	R/W	0h	This bit decides if the reset input is enabled or not. Setting this to 1 will cause the counter to reset to zero whenever the selected reset input goes active high. No event will be generated when the counter is reset. Setting this bit to 0 will cause the counter to ignore the reset inputs. Reset type: ERAD_RESET
8	CNT_INP_SEL_EN	R/W	0h	0 = Disable using the input_select register for the count input. The counter will always count CPU cycles. 1 = Enable using the input_select register for the count input. The counter will count the event selected by the count input register. Reset type: ERAD_RESET
7	NMI_EN	R/W	0h	0 = Will assert ERAD interrupt 1 = Will assert NMI Reset type: ERAD_RESET

Table 17-20. SEC_CNTL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	INTERRUPT	R/W	0h	This bit decides whether the counter module will generate interrupt when count value matches the reference. Note that the event outputs will always be generated regardless of the state of this bit. 0 SEC unit will not cause any action towards the CPU. 1 SEC unit will assert NMI/ERAD interrupt based on NMI_EN bit when the counter value matches the reference value. Reset type: ERAD_RESET
5	HALT	R/W	0h	This bit decides whether the counter module will generate a watchpoint to the CPU when the count value matches the reference. Note that the event outputs will always be generated regardless of the state of this bit. 0 SEC unit will not cause any action towards halting the CPU. 1 SEC unit will assert Halt request rest of the matching accesses(Watchpoint). These can cause the CPU to HALT Reset type: ERAD_RESET
4	RST_ON_MATCH	R/W	0h	This bit is used to decide whether the counter will reset to zero once it reaches the reference value. 0 Counter will stay at the reference value and the counter will go to COMPLETED state and further counting will be stopped. 1 The counter will reset to zero once it reaches the match value and will stay enabled. Reset type: ERAD_RESET
3	START_STOP_CUMULATIVE	R/W	0h	This bit decides whether the counter counts to give the cumulative cycle count for 'n' number of successive start stop events or clears the counter on very stop event to record the MAX_COUNT across successive start stop sequences. 0 When in START_STOP mode counter gets cleared on every stop event and MAX_COUNT records the max value 1 When in START_STOP mode counter keeps counting between successive start stop events to generate a cumulative count w/o clearing the counter on any stop events. MAX_COUNT register is invalid when this bit is set. Reset type: ERAD_RESET
2	START_STOP_MODE	R/W	0h	This bit is used to decide whether the counter will count in the START_STOP mode or not. 0 Normal count mode. The counter will not depend on the START and STOP events 1 This is the START-STOP mode of the counter. The counter will start counting only after the START input has been asserted. It will continue to count the selected event till the STOP event is seen. Reset type: ERAD_RESET
1	EDGE_LEVEL	R/W	0h	This bit is used to decide whether the counter will count the level of the event or the edge of the event. 0 Counter will increment the count as long as the count input is active high. 1 The counter will count only on the rising edge of the count input. Note: If the selected counter input signal was already high when SEC{#}_INPUT_SEL1.CNT_INP_SEL gets configured, the counter will increment by 1 when enabled, even if edge counting mode is selected Reset type: ERAD_RESET
0	EN	R/W	0h	0 = SEC Disabled 1 = SEC Enabled Reset type: ERAD_RESET

17.8.1.1.14 SEC_STATUS_j Register (Offset = 448h + formula) [Reset = 00000000h]

SEC_STATUS_j is shown in Figure 17-22 and described in Table 17-21.

Return to the Summary Table.

SEC Status Register

Offset = 448h + (j * 40h); where j = 0h to 3h

Figure 17-22. SEC_STATUS_j Register

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

Table 17-21. SEC_STATUS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	Reserved
9-8	STATUS	R	0h	SEC unit status 00 Idle 10 Enabled 11 Completed Reset type: ERAD_RESET
7-2	RESERVED	R	0h	Reserved
1	OVERFLOW	R	0h	This is a sticky bit which gets set every time the counter overflows and wraps around after reaching 0xffffffff. This bit will get cleared by writing a '1' to OVERFLOW of SEC_STATUSCLEAR register. Note : If hardware is trying to set OVERFLOW bit at the same cycle software is trying to clear, then the clear will be ignored and status will remain set Reset type: ERAD_RESET
0	EVENT_FIRED	R	0h	This is a sticky bit which gets set every time the SEC unit generates a match event. This will be used by software to figure out whether this SEC module fired an event or not. This bit will get cleared by writing a '1' to EVENT of SEC_STATUSCLEAR register. Note : If hardware is trying to set EVENT_FIRED bit at the same cycle software is trying to clear, then the clear will be ignored and status will remain set Reset type: ERAD_RESET

17.8.1.1.15 SEC_STATUSCLEAR_j Register (Offset = 44Ch + formula) [Reset = 00000000h]

SEC_STATUSCLEAR_j is shown in Figure 17-23 and described in Table 17-22.

Return to the Summary Table.

SEC Clear Register

Offset = 44Ch + (j * 40h); where j = 0h to 3h

Figure 17-23. SEC_STATUSCLEAR_j Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						OVERFLOW	EVENT_FIRED
R-0h						W1C-0h	W1C-0h

Table 17-22. SEC_STATUSCLEAR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	OVERFLOW	W1C	0h	Clear OVERFLOW: 0 No action. 1 A write with this bit set to 1 will clear the sticky OVERFLOW bit in the SEC_STATUS register. Reads of this bit position will always return a 0. Reset type: ERAD_RESET
0	EVENT_FIRED	W1C	0h	Clear EVENT_FIRED: 0 No action. 1 A write with this bit set to 1 will clear the sticky EVENT_FIRED bit in the SEC_STATUS register and bring the Breakpoint Module statemachine status back to ENABLED state if SEC is enable or IDLE state if SEC is disabled. Reads of this bit position will always return a 0. Reset type: ERAD_RESET

17.8.1.1.16 SEC_REF_j Register (Offset = 450h + formula) [Reset = FFFFFFFFh]

SEC_REF_j is shown in Figure 17-24 and described in Table 17-23.

Return to the Summary Table.

SEC Reference Register

Offset = 450h + (j * 40h); where j = 0h to 3h

Figure 17-24. SEC_REF_j Register

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

Table 17-23. SEC_REF_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REF	R/W	FFFFFFFFh	This register contains the counter reference value for comparison. The counter will generate an event if the count value matches the reference register. This register is writable by CPU only if application owns the unit, the writes are ignored. The register is writable by the debugger only if the debugger owns this unit. Otherwise, the writes are ignored. Reset type: ERAD_RESET

17.8.1.1.17 SEC_INPUT_SEL1_j Register (Offset = 454h + formula) [Reset = 00000000h]

SEC_INPUT_SEL1_j is shown in Figure 17-25 and described in Table 17-24.

Return to the Summary Table.

SEC Input Select Register1

Offset = 454h + (j * 40h); where j = 0h to 3h

Figure 17-25. SEC_INPUT_SEL1_j Register

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

Table 17-24. SEC_INPUT_SEL1_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	RST_INP_SEL	R/W	0h	These bits decide are used to select the event input that will be used as the reset input. These bits matter only if the Enable Reset bit is set to 1. Reset type: ERAD_RESET
15-8	RESERVED	R	0h	Reserved
7-0	CNT_INP_SEL	R/W	0h	These bits decide which of the inputs will be selected to enable counting. These inputs will be hooked up to the event outputs from the breakpoint module, counter module and to other system events Reset type: ERAD_RESET

17.8.1.1.18 SEC_INPUT_SEL2_j Register (Offset = 458h + formula) [Reset = 00000000h]

SEC_INPUT_SEL2_j is shown in Figure 17-26 and described in Table 17-25.

Return to the Summary Table.

SEC Input Select Register2

Offset = 458h + (j * 40h); where j = 0h to 3h

Figure 17-26. SEC_INPUT_SEL2_j Register

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

Table 17-25. SEC_INPUT_SEL2_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	STO_INP_SEL	R/W	0h	These bits decide which of the inputs will be selected as the STOP event for the counter. These inputs will be hooked up to the event outputs from the breakpoint module, counter module and to other system events. The usage of these bits are relevant only in the START_STOP mode of counting. Reset type: ERAD_RESET
15-8	RESERVED	R	0h	Reserved
7-0	STA_INP_SEL	R/W	0h	These bits decide which of the inputs will be selected as the START event for the counter. These inputs will be hooked up to the event outputs from the breakpoint module, counter module and to other system events. The usage of these bits are relevant only in the START_STOP mode of counting. Reset type: ERAD_RESET

17.8.1.1.19 SEC_INPUT_COND_j Register (Offset = 45Ch + formula) [Reset = 00000000h]

SEC_INPUT_COND_j is shown in Figure 17-27 and described in Table 17-26.

Return to the Summary Table.

SEC Input Conditioning Register

Offset = 45Ch + (j * 40h); where j = 0h to 3h

Figure 17-27. SEC_INPUT_COND_j Register

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

Table 17-26. SEC_INPUT_COND_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RST_INP_INV	R/W	0h	Invert the Selected Reset input Reset type: ERAD_RESET
11-10	RESERVED	R	0h	Reserved
9	RESERVED	R/W	0h	Reserved
8	STO_INP_INV	R/W	0h	Invert the Selected Stop input Reset type: ERAD_RESET
7-6	RESERVED	R	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	STA_INP_INV	R/W	0h	Invert the Selected Start input Reset type: ERAD_RESET
3-2	RESERVED	R	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	SEC_INP_INV	R/W	0h	Invert the Selected Counter input Reset type: ERAD_RESET

17.8.1.1.20 SEC_COUNT_j Register (Offset = 460h + formula) [Reset = 00000000h]

SEC_COUNT_j is shown in Figure 17-28 and described in Table 17-27.

Return to the Summary Table.

SEC Counter Register

Offset = 460h + (j * 40h); where j = 0h to 3h

Figure 17-28. SEC_COUNT_j Register

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

Table 17-27. SEC_COUNT_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	This register contains the current count value. The counter will generate an event if the count value matches the reference register (considering both upper and lower half of the register). This register is writable by CPU only if application owns the unit. Otherwise, the writes are ignored. The register is writable by the debugger only if the debugger owns this unit. Otherwise, the writes are ignored. NOTE : CPU and Debugger writes based on ownership are given for clearing these registers for fresh start's. If attempted when hardware is also updating, then register writes are given lower priority. Reset type: ERAD_RESET

17.8.1.1.21 SEC_MAX_COUNT_j Register (Offset = 464h + formula) [Reset = 00000000h]

SEC_MAX_COUNT_j is shown in Figure 17-29 and described in Table 17-28.

Return to the Summary Table.

SEC Max Count Register

Offset = 464h + (j * 40h); where j = 0h to 3h

Figure 17-29. SEC_MAX_COUNT_j Register

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

Table 17-28. SEC_MAX_COUNT_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MAX_COUNT	R/W	0h	This register contains the maximum recorded counter value. This is relevant only in the Start Stop mode of operation. NOTE : CPU and Debugger writes based on ownership are given for clearing these registers for fresh start's. If attempted when hardware is also updating, then register writes are given lower priority. Reset type: ERAD_RESET

17.8.1.1.22 SEC_MIN_COUNT_j Register (Offset = 468h + formula) [Reset = FFFFFFFFh]

SEC_MIN_COUNT_j is shown in Figure 17-30 and described in Table 17-29.

Return to the Summary Table.

SEC Min Count Register

Offset = 468h + (j * 40h); where j = 0h to 3h

Figure 17-30. SEC_MIN_COUNT_j Register

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

Table 17-29. SEC_MIN_COUNT_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MIN_COUNT	R/W	FFFFFFFFh	This register contains the minimum recorded counter value. This is relevant only in the Start Stop mode of operation. NOTE : CPU and Debugger writes based on ownership are given for clearing these registers for fresh start's. If attempted when hardware is also updating, then register writes are given lower priority. Reset type: ERAD_RESET

17.8.1.1.23 AND_MASK_OWNER_j Register (Offset = 640h + formula) [Reset = 00000000h]

AND_MASK_OWNER_j is shown in Figure 17-31 and described in Table 17-30.

Return to the Summary Table.

AND Owner Register

Offset = 640h + (j * 20h); where j = 0h to 3h

Figure 17-31. AND_MASK_OWNER_j Register

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

Table 17-30. AND_MASK_OWNER_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	Reserved
16	SEM	R	0h	1 => Access and owner from same zone (Additionally if SROOT LINK owns the block and access originated from SROOTLINK area) 0 => Access and owner from different zone (Additionally if SROOT LINK owns the block, but access originated from a non SROOT LINK area) These bits are only valid when OWNER = 10 (App owned) Reset type: ERAD_RESET
15-13	RESERVED	R	0h	Reserved
12	SROOT	R	0h	0 => This block is not owned by SROOT LINK 1 => This block is owned by SROOT LINK Reset type: ERAD_RESET
11-8	ZONE	R	0h	0000 => Zone-0 0001 => Zone-1 0010 => Zone-2 0011 => Zone-3 All others => Reserved These bits are only valid when OWNER = 10 (App owned) Reset type: ERAD_RESET
7-2	RESERVED	R	0h	Reserved

Table 17-30. AND_MASK_OWNER_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	OWNER	R/W	0h	00 => No Owner 01 => Debug Owned 10 => App Owned 11 => Reserved Debugger : Read-> Returns current ownership ID Writes -> Allowed only if current value is 00 or 01 Application: Read-> Returns current ownership ID Writes -> Allowed only if current value is 00 or 10 NOTE : When a 00 gets successfully written, all the rest of control/ status registers gets reset/cleared Reset type: ERAD_RESET

17.8.1.1.24 AND_MASK_CTL_j Register (Offset = 644h + formula) [Reset = 00000000h]

AND_MASK_CTL_j is shown in Figure 17-32 and described in Table 17-31.

Return to the Summary Table.

AND Control Register

Offset = 644h + (j * 20h); where j = 0h to 3h

Figure 17-32. AND_MASK_CTL_j Register

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

Table 17-31. AND_MASK_CTL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	NMI_EN	R/W	0h	0 = Will assert ERAD interrupt 1 = Will assert NMI Reset type: ERAD_RESET
1	INTERRUPT	R/W	0h	This bit decides whether this unit will generate interrupt when event matches occur. Note that the event outputs will always be generated regardless of the state of this bit. 0 This unit will not cause any action towards the CPU. 1 This unit will assert NMI/ERAD interrupt based on NMI_EN bit Reset type: ERAD_RESET
0	HALT	R/W	0h	This bit decides whether this unit will generate CPU halting signals when event matches occur. Note that the event outputs will always be generated regardless of the state of this bit. 0 This unit will not cause any action towards halting the CPU. 1 This unit will assert Halt request rest of the matching accesses(Watchpoint). These can cause the CPU to HALT Reset type: ERAD_RESET

17.8.1.1.25 EVENT_AND_MASK_j Register (Offset = 648h + formula) [Reset = 00FFFFFFh]

EVENT_AND_MASK_j is shown in Figure 17-33 and described in Table 17-32.

Return to the Summary Table.

AND Event Selection Register

Offset = 648h + (j * 20h); where j = 0h to 3h

Figure 17-33. EVENT_AND_MASK_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								MASK_EBC																							
R-0h								R/W-00FFFFFFh																							

Table 17-32. EVENT_AND_MASK_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	MASK_EBC	R/W	00FFFFFFh	AND event mask 0 Corresponding EVENT is enabled for EBC_EVENT_AND1 output 1 Corresponding EVENT is masked for EBC_EVENT_AND1 output Bit-0 EBC1 Bit-1 EBC2 .... Bit-15 EBC15 Bit-16 SEC1 Bit-17 SEC2 .... Bit-(23) SEC8 Reset type: ERAD_RESET

17.8.1.1.26 OR_MASK_OWNER_j Register (Offset = 740h + formula) [Reset = 00000000h]

OR_MASK_OWNER_j is shown in Figure 17-34 and described in Table 17-33.

Return to the Summary Table.

OR Owner Register

Offset = 740h + (j * 20h); where j = 0h to 3h

Figure 17-34. OR_MASK_OWNER_j Register

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

Table 17-33. OR_MASK_OWNER_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	Reserved
16	SEM	R	0h	1 => Access and owner from same zone (Additionally if SROOT LINK owns the block and access originated from SROOTLINK area) 0 => Access and owner from different zone (Additionally if SROOT LINK owns the block, but access originated from a non SROOT LINK area) These bits are only valid when OWNER = 10 (App owned) Reset type: ERAD_RESET
15-13	RESERVED	R	0h	Reserved
12	SROOT	R	0h	0 => This block is not owned by SROOT LINK 1 => This block is owned by SROOT LINK Reset type: ERAD_RESET
11-8	ZONE	R	0h	0000 => Zone-0 0001 => Zone-1 0010 => Zone-2 0011 => Zone-3 All others => Reserved These bits are only valid when OWNER = 10 (App owned) Reset type: ERAD_RESET
7-2	RESERVED	R	0h	Reserved

Table 17-33. OR_MASK_OWNER_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	OWNER	R/W	0h	00 => No Owner 01 => Debug Owned 10 => App Owned 11 => Reserved Debugger : Read-> Returns current ownership ID Writes -> Allowed only if current value is 00 or 01 Application: Read-> Returns current ownership ID Writes -> Allowed only if current value is 00 or 10 NOTE : When a 00 gets successfully written, all the rest of control/ status registers gets reset/cleared Reset type: ERAD_RESET

17.8.1.1.27 OR_MASK_CTL_j Register (Offset = 744h + formula) [Reset = 00000000h]

OR_MASK_CTL_j is shown in Figure 17-35 and described in Table 17-34.

Return to the Summary Table.

OR Control Register

Offset = 744h + (j * 20h); where j = 0h to 3h

Figure 17-35. OR_MASK_CTL_j Register

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

Table 17-34. OR_MASK_CTL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	NMI_EN	R/W	0h	0 = Will assert ERAD interrupt 1 = Will assert NMI Reset type: ERAD_RESET
1	INTERRUPT	R/W	0h	This bit decides whether this unit will generate interrupt when event matches occur. Note that the event outputs will always be generated regardless of the state of this bit. 0 This unit will not cause any action towards the CPU. 1 This unit will assert NMI/ERAD interrupt based on NMI_EN bit Reset type: ERAD_RESET
0	HALT	R/W	0h	This bit decides whether this unit will generate CPU halting signals when event matches occur. Note that the event outputs will always be generated regardless of the state of this bit. 0 This unit will not cause any action towards halting the CPU. 1 This unit will assert Halt request rest of the matching accesses(Watchpoint). These can cause the CPU to HALT Reset type: ERAD_RESET

17.8.1.1.28 EVENT_OR_MASK_j Register (Offset = 748h + formula) [Reset = 00FFFFFFh]

EVENT_OR_MASK_j is shown in Figure 17-36 and described in Table 17-35.

Return to the Summary Table.

OR Event Selection Register

Offset = 748h + (j * 20h); where j = 0h to 3h

Figure 17-36. EVENT_OR_MASK_j Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								MASK_EBC																							
R-0h								R/W-00FFFFFFh																							

Table 17-35. EVENT_OR_MASK_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	MASK_EBC	R/W	00FFFFFFh	OR event mask 0 Corresponding EVENT is enabled for EBC_EVENT_OR1 output 1 Corresponding EVENT is masked for EBC_EVENT_OR1 output Bit-0 EBC1 Bit-1 EBC2 .... Bit-15 EBC15 Bit-16 SEC1 Bit-17 SEC2 .... Bit-(23) SECn Reset type: ERAD_RESET

17.8.1.1.29 PCTRACE_OWNER Register (Offset = 840h) [Reset = 00000000h]

PCTRACE_OWNER is shown in Figure 17-37 and described in Table 17-36.

Return to the Summary Table.

Owner Register

Figure 17-37. PCTRACE_OWNER Register

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

Table 17-36. PCTRACE_OWNER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	Reserved
16	SEM	R	0h	1 => Access and owner from same zone (Additionally if SROOT LINK owns the block and access originated from SROOTLINK area) 0 => Access and owner from different zone (Additionally if SROOT LINK owns the block, but access originated from a non SROOT LINK area) These bits are only valid when OWNER = 10 (App owned) Reset type: ERAD_RESET
15-13	RESERVED	R	0h	Reserved
12	SROOT	R	0h	0 => This block is not owned by SROOT LINK 1 => This block is owned by SROOT LINK Reset type: ERAD_RESET
11-8	ZONE	R	0h	0000 => Zone-0 0001 => Zone-1 0010 => Zone-2 0011 => Zone-3 All others => Reserved These bits are only valid when OWNER = 10 (App owned) Reset type: ERAD_RESET
7-2	RESERVED	R	0h	Reserved

Table 17-36. PCTRACE_OWNER Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	OWNER	R/W	0h	00 => No Owner 01 => Debug Owned 10 => App Owned 11 => Reserved Debugger : Read-> Returns current ownership ID Writes -> Allowed only if current value is 00 or 01 Application: Read-> Returns current ownership ID Writes -> Allowed only if current value is 00 or 10 NOTE : When a 00 gets successfully written, all the rest of control/ status registers gets reset/cleared Reset type: ERAD_RESET

17.8.1.1.30 PCTRACE_GLOBAL Register (Offset = 844h) [Reset = 00000000h]

PCTRACE_GLOBAL is shown in Figure 17-38 and described in Table 17-37.

Return to the Summary Table.

Global Control Register

Figure 17-38. PCTRACE_GLOBAL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						BUFFER_SIZE	
R-0h						R-0h	
15	14	13	12	11	10	9	8
RESERVED							INIT
R-0h							W1C-0h
7	6	5	4	3	2	1	0
RESERVED							EN
R-0h							R/W-0h

Table 17-37. PCTRACE_GLOBAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17-16	BUFFER_SIZE	R	0h	Trace buffer size 00 = 1 Kbyte 01 = 2 kbyte 10 = 3 kbyte 11 = 4 kbyte Reset type: ERAD_RESET
15-9	RESERVED	R	0h	Reserved
8	INIT	W1C	0h	0 = No action 1 = Trace module is initialized for a fresh trace start with buffer pointer(PTR) reset and BUFFER_FULL flags cleared along with PC_SOFTENABLE and PC_SOFTDISABLE Reads of this bit position always returns zero Reset type: ERAD_RESET
7-1	RESERVED	R	0h	Reserved
0	EN	R/W	0h	0 = PC Trace Disabled 1 = PC Trace Enabled Reset type: ERAD_RESET

17.8.1.1.31 PCTRACE_BUFFER Register (Offset = 848h) [Reset = 00000000h]

PCTRACE_BUFFER is shown in Figure 17-39 and described in Table 17-38.

Return to the Summary Table.

Trace Buffer pointer register

Figure 17-39. PCTRACE_BUFFER Register

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

Table 17-38. PCTRACE_BUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	Reserved
16	BUFFER_FULL	R	0h	0 = Trace Buffer Never became full/Overflowed after init 1 = Indicates Trace Buffer became full/Overflowed This bit also gets cleared when INIT is performed via PCTRACE_GLOBAL.INIT Reset type: ERAD_RESET
15-0	PTR	R	0h	Current Pointer to the Trace Buffer. If PTR=0 => there is no trace data in buffer (when BUFFER_FULL=0) If PTR=2,4,6. there is fresh trace data in buffer Buffer pointer gets incremented by 2 for every new trace storage, since two 32-bit values get stored for every discontinuity. For ex : 2 => Locations 0,1 of trace buffer have valid data (SRC,DST) 4 => Locations 0,1,2,3 have valid data (SRC,DST,SRC,DST) and so on Thhese bits also gets cleared when INIT is performed via PCTRACE_GLOBAL.INIT Reset type: ERAD_RESET

17.8.1.1.32 PCTRACE_QUAL1 Register (Offset = 84Ch) [Reset = 00000000h]

PCTRACE_QUAL1 is shown in Figure 17-40 and described in Table 17-39.

Return to the Summary Table.

Trace Qualifier register 1

Figure 17-40. PCTRACE_QUAL1 Register

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

Table 17-39. PCTRACE_QUAL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	STOP_INP_INV	R/W	0h	Invert the Selected Stop input Reset type: ERAD_RESET
21	RESERVED	R/W	0h	Reserved
20	START_INP_INV	R/W	0h	Invert the Selected Start input Reset type: ERAD_RESET
19	RESERVED	R/W	0h	Reserved
18	WINDOWED_INP_INV	R/W	0h	Invert the Selected trace input Reset type: ERAD_RESET
17-16	TRACE_MODE	R/W	0h	0x = Trace without any hardware qualifiers 10 = Trace using Windowed mode 11 = Trace using Start/Stop mode These two bits are valid only when PCTRACE_GLOBAL.EN is set to '1' Reset type: ERAD_RESET
15-8	RESERVED	R	0h	Reserved
7-0	WINDOWED_INP_SEL	R/W	0h	These bits decide which of the inputs will be selected to enable tracing.These inputs will be hooked up to the event outputs from the bus comparator module, counter module and other system events. The usage of these bits are relevant only in the Windowed mode of trace module. Pls refer to the device spec for complete list of signals Reset type: ERAD_RESET

17.8.1.1.33 PCTRACE_QUAL2 Register (Offset = 850h) [Reset = 00000000h]

PCTRACE_QUAL2 is shown in Figure 17-41 and described in Table 17-40.

Return to the Summary Table.

Trace Qualifier register 2

Figure 17-41. PCTRACE_QUAL2 Register

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

Table 17-40. PCTRACE_QUAL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	STOP_INP_SEL	R/W	0h	These bits decide which of the inputs will be selected as the STOP event for trace. These inputs will be hooked up to the event outputs from the bus comparator module, counter module and other system events. The usage of these bits are relevant only in the Start/Stop mode of trace module. Pls refer to the device spec for complete list of signals Reset type: ERAD_RESET
15-8	RESERVED	R	0h	Reserved
7-0	START_INP_SEL	R/W	0h	These bits decide which of the inputs will be selected as the START event for trace. These inputs will be hooked up to the event outputs from the bus comparator module, counter module and other system events. The usage of these bits are relevant only in the Start/Stop mode of trace module. Pls refer to the device spec for complete list of signals Reset type: ERAD_RESET

17.8.1.1.34 PCTRACE_LOGPC_SOFTENABLE Register (Offset = 854h) [Reset = 00000000h]

PCTRACE_LOGPC_SOFTENABLE is shown in Figure 17-42 and described in Table 17-41.

Return to the Summary Table.

PC when PC Trace was last enabled by software

Figure 17-42. PCTRACE_LOGPC_SOFTENABLE Register

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

Table 17-41. PCTRACE_LOGPC_SOFTENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PC_SOFTENABLE	R	0h	These bits reflect the value of PC when trace module enable bit was last written with '1' (PCTRACE_GLOBAL.EN).These registers are primarily used by ccs drivers while displaying the trace to give a logical start from where tracing was enabled. This register also gets cleared when INIT is performed via PCTRACE_GLOBAL.INIT Reset type: ERAD_RESET

17.8.1.1.35 PCTRACE_LOGPC_SOFTDISABLE Register (Offset = 858h) [Reset = 00000000h]

PCTRACE_LOGPC_SOFTDISABLE is shown in Figure 17-43 and described in Table 17-42.

Return to the Summary Table.

PC when PC Trace was last disabled by software

Figure 17-43. PCTRACE_LOGPC_SOFTDISABLE Register

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

Table 17-42. PCTRACE_LOGPC_SOFTDISABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PC_SOFTDISABLE	R	0h	These bits reflect the value of PC when trace module enable bit was last written with '0' (PCTRACE_GLOBAL.EN).These registers are primarily used by ccs drivers while displaying the trace to give a logical end to where tracing block was disabled. This register also gets cleared when INIT is performed via PCTRACE_GLOBAL.INIT Reset type: ERAD_RESET

17.8.1.1.36 PCTRACE_BUFFER_BASE_y Register (Offset = 1000h + formula) [Reset = 00000000h]

PCTRACE_BUFFER_BASE_y is shown in Figure 17-44 and described in Table 17-43.

Return to the Summary Table.

Trace Buffer Base address

Offset = 1000h + (y * 4h); where y = 0h to FFh

Figure 17-44. PCTRACE_BUFFER_BASE_y Register

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

Table 17-43. PCTRACE_BUFFER_BASE_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	PROGRAM_COUNTER	R	0h	Program Counter[31:1] value where discontinuity occurred Note: PC values are aligned for min 16-bit instruction size so bit 0 of program counter is always 0. 32 bit PC value will be {PCTRACE_BUFFER_BASE.PROGRAM_COUNTER,1'b0} Reset type: ERAD_RESET
0	BLOCKED	R	0h	1 = PROGRAM_COUNTER[31:1] is not valid due to security permissions 0 = PROGRAM_COUNTER[31:1] is valid Reset type: ERAD_RESET

Chapter 18

Data Logger and Trace (DLT)

This chapter describes the features and operation of the data logger and trace (DLT) module. The DLT module enhances real-time non-intrusive data-logging and system analysis capabilities of the device. Filtering out what is data-logged, using generated ERAD start commands to initiate data-logging and stop commands, using DLT start and stop commands for data-logging, and time stamping when or where in the program code data is being logged are the main features of the DLT. The DLT is accessible by the RTDMA and CPU.

18.1 Introduction

For critical CPU run-time content the data logger and trace (DLT) module has the ability to control what data gets logged, when to start data-logging, and the size of the data to capture. Critical run-time content can include any information that needs to be monitored as the content is computed. When data-logging the DLT is non-intrusive meaning there is no impact to run-time or CPU core behavior. The ability to view intermediate values of computation in a critical task, such as a control loop, can help users fine-tune the loop. The DLT module can generate interrupts to the interrupt controller, issue RTDMA transfer requests, and interact with ERAD event triggers.

The DLT can collect, time-stamp, prefilter, export, and do real-time and post analysis of data.

18.1.1 Features

The DLT has the following capabilities:

- Logging critical run-time content referred to as data logging
- Analyze program execution sequence using tags referred to as trace
- Logging is non-intrusive to run-time/CPU core behavior
- Flexible logging capability for extended period of time by transferring data to external memory or short period of time to on-chip memory
- Logging of registers can be up to 32-bit size, depending on the size on the variable to be logged
- Each CPU has DLT support
- Time stamping records time difference from last logged variable and can time stamp the IPC timer's count
- RTDMA triggering
- Global, FIFO and timer interrupt generation

18.1.2 DLT Related Collateral

Foundational Materials

- C29x Academy - Data Log and Trace (DLT)

18.1.3 Interfaces

The DLT has the following interfaces:

- System Clock
- System Reset
- CPU
- ERAD
- IPC Timer
- Peripheral data bus
- RTDMA
- CPU interrupt flags

18.1.3.1 Block Diagram

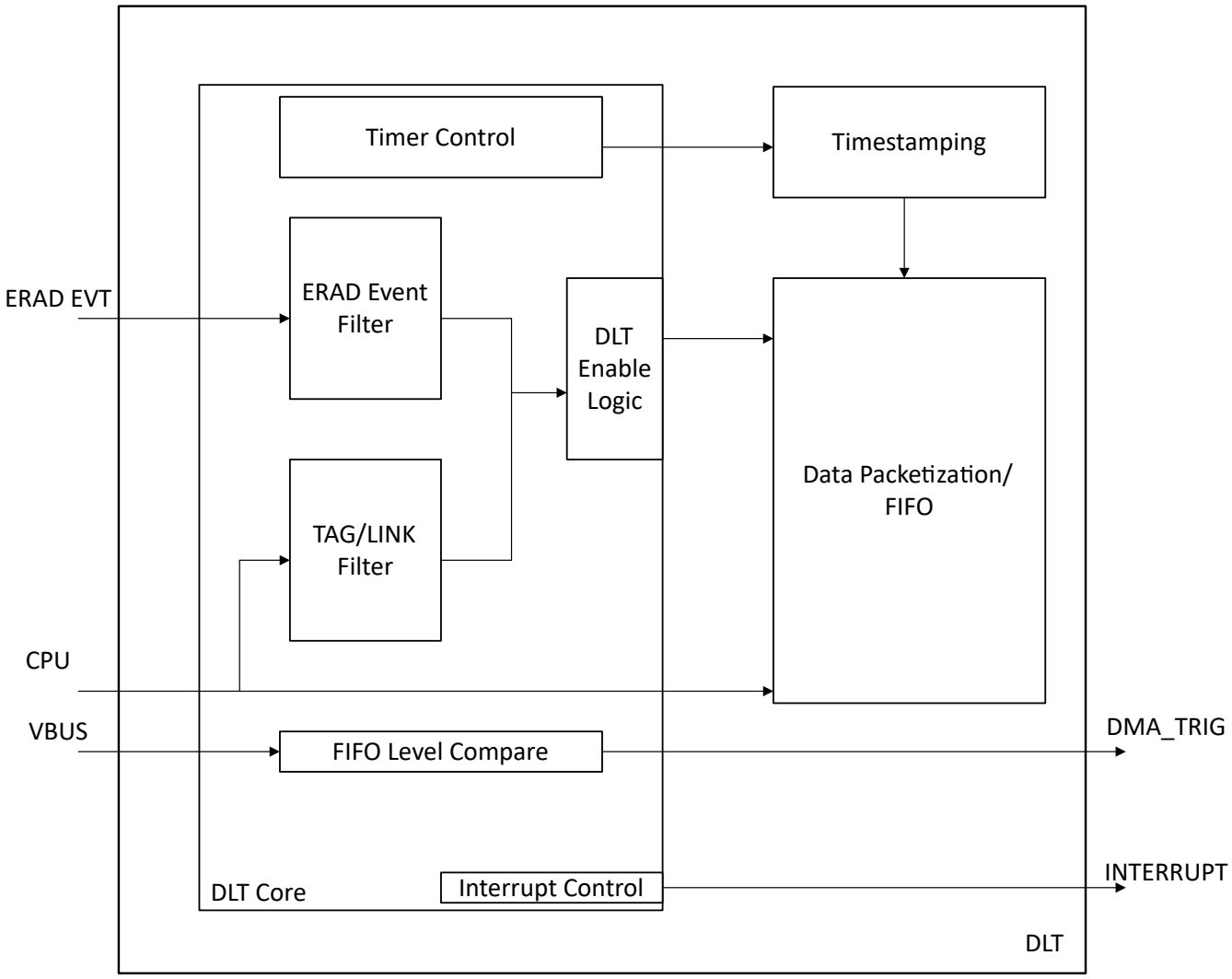


Figure 18-1. DLT Block Diagram

18.2 Functional Overview

The DLT operates mainly between the CPU and RTDMA. The DLT monitors data logging information coming from the CPU, applies filters on what data to log and stores the data in the local FIFO. When enough data is collected, the DLT requests a CPU Interrupt and a RTDMA transfer request, either of which must take contents out of the FIFO and store the contents on on-chip RAM/External RAM or export the contents out using the FSI for example. The user can choose to enable either the RTDMA or CPU Interrupt to perform this task.

18.2.1 DLT Configuration

DLT module can be configured with the steps as followed:

1. What capture mode to configure the DLT?
   a. With a TAG or DLREG entry, the mode can be configured to capture the program counter or time of the event.
2. When did an entry occur? (Only when capture mode is set to track time of event)
   a. Time-stamping provides the CPU the time when data logging started and when the DLREG has stored value.
3. Where to start and end data logging? (TAG Filter)
   a. TAG filters help with indicating to the CPU where to start and end data logging.
   b. In addition to TAG filters, ERAD start/end filters can also be used.
4. How to export the data log? (FIFO construction)
   a. Internal on-chip memory
   b. External memory
5. What code region is enabled for data log? (LINK Filter)

The filtering unit provides security configurations to specify what links can be available for data-logging before production and during development. The filtering unit can regulate throughput from DLT FIFO buffer by attaching a tag to a variable or register to capture the value for data logging.

The DLT_EN configuration acts as a global enable for DLT, whereby all components of the DLT module can be completely disabled. All configurations of DLT discussed in upcoming sections are part of DLT_CORE_REGS and are only configurable from CPUx.LINK2.

The data bus is used to access registers and FIFO memory in DLT. DLT configurations are located under DLT_CORE_REGS. The internal FIFO is available in both FIFO mode (FIFO_REGS), and memory mode (FIFO_MEM) for debug.

If there is an incorrect access by CPU, error is generated, and access is dropped. In case of incorrect debugger access, no error is generated but access is dropped. The below table describes the access protections and permissions for the registers belonging to DLT.

Table 18-1. DLT Data Bus Apertures

Register Map	CPU Access Protection	Debugger Access Permissions	DMA Access Permission
DLT_CORE_REGS	Only LINK2 of corresponding CPU can write/read	Zone corresponding to LINK2, if enabled for debug can write/read	No Access
FIFO_REGS	LINKs of corresponding CPU can write/read	ZONE corresponding to LINK enabled by SSU, is enabled for debug read No write access	LINKs enabled by DMA can read. No write accesss
FIFO_MEM	Not writable	ZONE corresponding to LINK enabled by SSU, is enabled for debug read. No write access	LINKs enabled by DMA can read. No write access

Note

Access protections/permissions implemented inside SSU. DMA access configurations implemented in DMA.

18.2.1.1 LINK Filter

The purpose of this filter is security. To prevent data being logged by one user, that is, one LINK, from being visible to another, a LINK qualifier is sent with each CPI (Co-processor interface) access from the CPU. DLT has a bit to enable/disable each LINK'ss DLT entries from being captured into the internal FIFO in the LINK_EN register. This helps control visibility of sensitive data.

For example, if data logging all sections of code belonging to a CPU is required then enabling all LINKs for the CPU would be the correct configuration.

This is also a feature that allows different data logging controls during development time versus production, as data logging is useful during both. Application code may want to disable data logging of certain regions before production.

In both cases, DLT code remains the same and the DLT instructions stay embedded in the user code.

18.2.1.2 TAG Filter

The purpose of this filter is to regulate throughput via the DLT FIFO, and to divide Data Logging within a LINK into multiple sections.

Each Data Log set is preceded by a TAG identifier. This TAG can be masked with a reference and used to enable/disable entries into the internal FIFO. This also helps in preventing overflow of Data Logging FIFO, by prioritizing the different Data Logging sections. This feature must be enabled using TAG_FILTER_EN, which is a static configuration.

If TAG_FILTER_EN && ((TAG & TAG_FILTER_START_MASK) = TAG_FILTER_START_REFERENCE), data logging is started.

If TAG_FILTER_EN && ((TAG & TAG_FILTER_END_MASK) = TAG_FILTER_END_REFERENCE), data logging is stopped.

18.2.1.3 ERAD Event Trigger

The purpose of this filter is to regulate throughput via the DLT FIFO, via Hardware events. ERAD generates SEC and EBC events, derived from processing system events.

DLT has a bit for each ERAD input to DLT, separately for START and STOP, and a bit to enable the ERAD filtering logic, ERAD_FILTER_EN. By default, if LINK and TAG filters are enabled, Data is Logged. When ERAD_FILTER_EN is high, if ERAD stop event is triggered, Data Logging is paused until the next ERAD start event.

Table 18-2. ERAD START and END MASK Lower 32 Events

ERAD_START/END_MASK_L	ERAD Event
0	CPUx_ERAD_EBC_EVT1
1	CPUx_ERAD_EBC_EVT2
2	CPUx_ERAD_EBC_EVT3
3	CPUx_ERAD_EBC_EVT4
4	CPUx_ERAD_EBC_EVT5
5	CPUx_ERAD_EBC_EVT6
6	CPUx_ERAD_EBC_EVT7
7	CPUx_ERAD_EBC_EVT8
8-15	Reserved
16	CPUx_ERAD_SEC_EVT1
17	CPUx_ERAD_SEC_EVT2
18	CPUx_ERAD_SEC_EVT3
19	CPUx_ERAD_SEC_EVT4
20-23	Reserved

Table 18-2. ERAD START and END MASK Lower 32 Events (continued)

ERAD_START/END_MASK_L	ERAD Event
24	CPUx_ERAD_AND_MASK1
25	CPUx_ERAD_AND_MASK2
26	CPUx_ERAD_AND_MASK3
27	CPUx_ERAD_AND_MASK4
28-31	Reserved

Table 18-3. ERAD START and END MASK Higher 32 Events

ERAD_START/END_MASK_H	ERAD Event
0	CPUx_ERAD_OR_MASK1
1	CPUx_ERAD_OR_MASK2
2	CPUx_ERAD_OR_MASK3
3	CPUx_ERAD_OR_MASK4
4-31	Reserved

18.2.1.4 Concurrent FILTERING modes

LINK FILTER is a necessary condition for data logging in FIFO. If a LINK is masked, data logging is disabled irrespective of other filters. However, in case of any overlap between TAG FILTER and ERAD FILTER, or if both TAG/ERAD START/STOP events are triggered at the same time, the START command takes precedence and data logging is allowed.

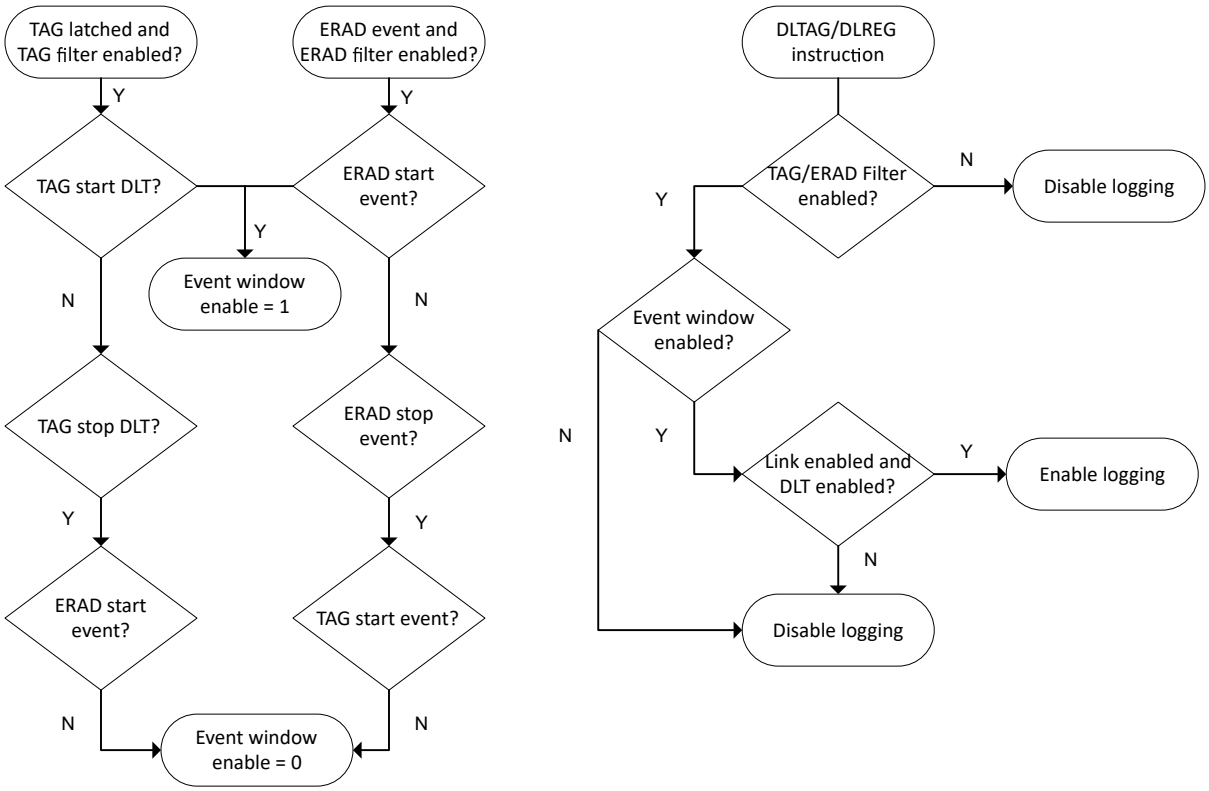


Figure 18-2. Data Log Decision Chart

If a START event from either TAG or ERAD is received, a windowed signal is set and cleared when a STOP event occurs from either TAG or ERAD. When repeated START events without a STOP event occur, the repeated start events have no meaning.

Filter enables are not expected to change dynamically in the middle of data logging, and take effect after a few cycles. The user must give sufficient time before inserting DLT instructions, while re-configuring.

Note

It is recommended to use a non zero reference TAG. In the case where TAG is a 0 and TAG reference is 0, and TAG filter is enabled, the DLT will have a false start condition.

18.2.2 Time-stamping

As part of DLT, time-stamping is done internally for every entry in the FIFO. This is a simple scheme of continuously running counters, and not clock time. Since the structure of DLT code is such that every data set comes with TAG identifier, two counters are used, one for TAGs and the other for REGs. The time-stamping structure has been split to give higher granularity to the TAG inputs.

TIMER1 is exported with TAGS and is a 45 bit timer. TIMER1's value is derived from the 64-bit IPC timer outside DLT.

The TIMER2 is exported with REGS. This indicates the time-difference from the last reported TAG value. TIMER2 is an internal counter.

Also, from a use case perspective, the relative time-stamp of data (REGs) with respect to TAGs gives a clear indication of the code performance for the section. To compare across different sections of code, the absolute values from TAGs can be compared.

TIMER 2 has reset controls that stop the timer and resets the timer to 0. In addition, whenever a new TAG is received and passes through the filters, the TIMER1 overflow, TIMER2 overflow and TIMER2 counter are also reset, so that the relative value is now computed from the newly received TAG.

In case there is a data logging section that is interrupted and a new TAG is received, the relative time and TAG filtering of the subsequent entries is calculated based on Interrupt TAG. It is recommended to use TAG filtering at the start of every ISR or function call, followed by the registers to be data logged.

18.2.3 FIFO Construction

All DLT entries, both TAG and REG, that have passed the FILTERs are recorded in the FIFO. The FIFO is constructed as 64 bit rows, with the REG/TAG, TIMESTAMP/PC source and Overflow Indicators.

Each packet received is entered on a new row. The C29x CPU and DLT both support TAG of size 8 bits and 16 bits. Always allocate 16 bits for the TAG in the FIFO irrespective of the incoming TAG size. The maximum size of REG supported in the C29x CPU is 64 bits, but only up-to 32 bits are supported in DLT. The 64-bit REG is not supported.

DLT Capture mode static configuration decides whether to capture TIMER/PC value. A packet does not differentiate between them and must be analyzed by the user based on the configuration set. The following tables depict how the raw data stored in the FIFO must be interpreted. The LSB represents whether the data log is a TAG or REG, where TAG = 1 and REG = 0.

Table 18-4. FIFO 16-bit TAG Structure

Higher 32 Bits (FIFO_BUF_H)		Lower 32 Bits (FIFO_BUF_L)			
TIMER1's MSBs	**16-bit TAG Entry (MSBs Padded with Zeros if 8-bit TAG)**	**TIMER1's LSBs**	**TIMER1's Overflow**	**FIFO Overflow**	**TAG**
16 bits [31:16]	16 bits [15:0]	29 bits [31:3]	1 bit [2]	1 bit [1]	1

Table 18-5. FIFO 16-bit PC TAG Structure

Higher 32 Bits (FIFO_BUF_H)		Lower 32 Bits (FIFO_BUF_L)		
Reserved	**TAG Entry**	**Program Counter**	**FIFO Overflow**	**TAG**
16 bits [31:16]	16 bits [15:0]	30 bits [31:2]	1 bit [1]	1

Table 18-6. FIFO 32-bit REG Structure

Higher 32 Bits (FIFO_BUF_H)	Lower 32 Bits (FIFO_BUF_L)			
REG Content	**TIMER2's Count**	**TIMER2's Overflow**	**FIFO Overflow**	**REG**
32 bits [63:32]	29 bits [31:3]	1 bit [2]	1 bit [1]	0

Table 18-7. FIFO 32-bit PC REG Structure

Higher 32 Bits (FIFO_BUF_H)	Lower 32 Bits (FIFO_BUF_L)		
REG Content	**Program Counter**	**FIFO Overflow**	**REG**
32 bits [63:32]	30 bits [31:2]	1 bit [1]	0

In C29x CPU, the minimum instruction is 16-bit size. Hence, the PC always points to a 16-bit word size, and the 0th bit of the PC is irrelevant. The EMIF space in the first device is in the upper 2GB, and is not covered, but you cannot execute code from EMIF space anyway. Hence, using PC[30:1] as a Trace source is used. Also note that in PC mode, there is no Time entry in FIFO. Hence, no TIMER overflow is indicated. To read the PC value properly, left shift by 1 yields the correct PC capture.

TAG/REG indication has been kept at bit[0] of the FIFO for ease of software processing. In assembly, the LSB can be ANDed with 1 and an if-else discontinuity can be executed. Moving it elsewhere requires code to an add right-shift operation that you want to avoid. Similarly, TAGs are also preferred to be aligned to 32-bit boundary for ease of software processing, since TAGs are decision making points. To enable this, DLTIME has been split up into lower and higher words, when Capture mode is TIMER.

18.2.3.1 FIFO Interrupt

The FIFO is implemented as a memory and the depth of FIFO is 256. Three interrupts are generated from the FIFO:

- When the FIFO is full and a write is attempted (overflow).
- When a read is attempted with no data in the FIFO (underflow).
- When the number of writes in the FIFO reaches the configured trigger level. This also initiates a RTDMA request, if enabled.

Reading a 1 from the global interrupt status within INT_FLG [INT] indicates that an interrupt was generated from the events.

The FIFO read pointer is incremented on a 32-bit read to FIFO_H register by any of the peripherals.

Cases where FIFO pointers are not incremented, and statuses are not modified:

- If only FIFO_L register is read
- If FIFO_H register is read using lesser than 32-bit access size
- If FIFO is read through FIFO_MEM aperture
- If FIFO is read using Debug Access

18.3 Software

18.3.1 DLT Registers to Driverlib Functions

Table 18-8. DLT Registers to Driverlib Functions

File	Driverlib Function
ERAD_START_MASK_L	
dlt.c	DLT_configERADFilter
ERAD_START_MASK_H	
dlt.c	DLT_configERADFilter
ERAD_END_MASK_L	
dlt.c	DLT_configERADFilter
ERAD_END_MASK_H	
dlt.c	DLT_configERADFilter
TAG_FILTER_START_REF	
dlt.c	DLT_configTagFilter
TAG_FILTER_START_MASK	
dlt.c	DLT_configTagFilter
TAG_FILTER_END_REF	
dlt.c	DLT_configTagFilter
TAG_FILTER_END_MASK	
dlt.c	DLT_configTagFilter
LINK_EN	
dlt.h	DLT_enableLinkPermission
dlt.h	DLT_disableLinkPermission
CONTROL	
dlt.h	DLT_enableModule
dlt.h	DLT_disableModule
dlt.h	DLT_enableERADFilter
dlt.h	DLT_disableERADFilter
dlt.h	DLT_enableTagFilter
dlt.h	DLT_disableTagFilter
dlt.h	DLT_setCaptureMode
dlt.h	DLT_resetFilter
FIFO_CONTROL	
dlt.h	DLT_enableDMA
dlt.h	DLT_disableDMA
dlt.h	DLT_resetFIFO
dlt.h	DLT_setFIFOTriggerLevel
TIMER_CONTROL	
dlt.h	DLT_resetTimer
FIFO_STS	
dlt.h	DLT_getFIFOWordStatus
dlt.h	DLT_getFIFOWriteStatus
FIFO_PTR	
dlt.h	DLT_getFIFOReadPointer
dlt.h	DLT_getFIFOWritePointer
TIMER2_COUNT	

Table 18-8. DLT Registers to Driverlib Functions (continued)

File	Driverlib Function
dlt.h	DLT_getTimerCount
INT_FLG	
dlt.h	DLT_getGlobalInterruptStatus
dlt.h	DLT_getTimer1OverflowStatus
dlt.h	DLT_getTimer2OverflowStatus
dlt.h	DLT_getFIFOOverflowStatus
dlt.h	DLT_getFIFOUnderflowStatus
dlt.h	DLT_getFIFOTriggerLevelStatus
INT_EN	
dlt.h	DLT_enableInterrupt
dlt.h	DLT_disableInterrupt
INT_FRC	
dlt.h	DLT_forceEvent
INT_CLR	
dlt.h	DLT_clearEvent
FIFO_BUF_L	
-	
FIFO_BUF_H	
-	
FIFO_MEM(i)	
-	

18.3.2 DLT Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/dlt

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

18.3.2.1 DLT TAG filter example - SINGLE_CORE

FILE: dlt_ex1_datalog_tag.c

This example configures DLT and datalogs four variables MultVar, Current, Speed, Voltage throughout code execution. The DLT contents are read from the DLTFIFO via CPU and transferred to internal memory.

- DLT configured for Time capture mode, tag filter enabled
- In order to use the tag filter, compiler intrinsics are used:
- __builtin_c29_datalog_tag for tag
- __builtin_c29_datalog_write for reg

18.3.2.2 DLT TAG filter example - SINGLE_CORE

FILE: dlt_ex2_dma_fsi_export.c

This example, configures the DLT to data log a single variable within a CPU Timer ISR and export out the DLT contents via FSI. The DMA is used to transfer the contents. When the FIFO reaches a level of 4, the DMA is triggered to transfer contents to FSI TX buffer.

DLT is setup using TAG filter and REGs in order to data log.

- DLT configured for Time capture mode, tag filter enabled
- In order to use the tag filter, compiler intrinsics are used:
- __builtin_c29_datalog_tag for tag

- __builtin_c29_datalog_write for reg

18.3.2.3 DLT ERAD filter example - SINGLE_CORE

FILE: dlt_ex3_datalog_erad.c

This example configures DLT and datalogs four variables MultVar, Current, Speed, Voltage throughout code execution. The DLT contents are read from the DLTFIFO via CPU and transferred to internal memory.

- DLT configured for Time capture mode, ERAD filter enabled
- In order to use the ERAD filter, ERAD is configured in SysCfg:

18.4 DLT Registers

This Section describes the DLT Registers.

18.4.1 DLT Base Address Table

Table 18-9. DLT Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
DLT_CORE_REGS	CPUDLT_BASE	0x3001_0000	-	YES	YES	YES	-	-	-	YES
DLT_FIFO_REGS	CPU1DLTFIFO_BASE	0x600F_8000	YES	YES	YES	YES	YES	YES	-	YES
DLT_FIFO_REGS	CPU2DLTFIFO_BASE	0x600F_A000	YES	YES	YES	YES	YES	YES	-	YES
DLT_FIFO_REGS	CPU3DLTFIFO_BASE	0x600F_C000	YES	YES	YES	YES	YES	YES	-	YES

18.4.2 DLT_CORE_REGS Registers

Table 18-10 lists the memory-mapped registers for the DLT_CORE_REGS registers. All register offset addresses not listed in Table 18-10 should be considered as reserved locations and the register contents should not be modified.

Table 18-10. DLT_CORE_REGS Registers

Offset	Acronym	Register Name	Protection
0h	ERAD_START_MASK_L	ERAD Start Mask for Lower 32 lines	
4h	ERAD_START_MASK_H	ERAD Start Mask for Higher 32 lines	
8h	ERAD_END_MASK_L	ERAD End Mask for Lower 32 lines	
Ch	ERAD_END_MASK_H	ERAD End Mask for Higher 32 lines	
10h	TAG_FILTER_START_REF	Tag Filer Start Reference	
14h	TAG_FILTER_START_MASK	Tag Filer Start Mask	
18h	TAG_FILTER_END_REF	Tag Filer End Reference	
1Ch	TAG_FILTER_END_MASK	Tag Filer End Mask	
20h	LINK_EN	Link Enable	
24h	DLT_CONTROL	DLT Control Register	KEY:KEY=0x5a5a
28h	FIFO_CONTROL	FIFO Control Register	
2Ch	TIMER_CONTROL	Timer Control Register	
30h	FIFO_STS	Number of entries in FIFO	
34h	FIFO_PTR	Pointer locations in FIFO	
38h	TIMER2_COUNT	Timer2 Status	
3Ch	INT_FLG	Interrupt Flag	
40h	INT_EN	Interrupt Enable	
44h	INT_FRC	Interrupt Force	
48h	INT_CLR	Interrupt Clear	

Complex bit access types are encoded to fit into small table cells. Table 18-11 shows the codes that are used for access types in this section.

Table 18-11. DLT_CORE_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.

Table 18-11. DLT_CORE_REGS Access Type Codes (continued)

Access Type	Code	Description
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

18.4.2.1 ERAD_START_MASK_L Register (Offset = 0h) [Reset = 00000000h]

ERAD_START_MASK_L is shown in Figure 18-3 and described in Table 18-12.

Return to the Summary Table.

ERAD Start Mask for Lower 32 lines

Figure 18-3. ERAD_START_MASK_L Register

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

Table 18-12. ERAD_START_MASK_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	START_MASK	R/W	0h	ERAD can generate upto 64 events per instance. This mask Is applied on the first 32 events. This mask if set generates a START DLT event, after which entries are recorded in the FIFO Reset type: SYSRSn

18.4.2.2 ERAD_START_MASK_H Register (Offset = 4h) [Reset = 00000000h]

ERAD_START_MASK_H is shown in Figure 18-4 and described in Table 18-13.

Return to the Summary Table.

ERAD Start Mask for Higher 32 lines

Figure 18-4. ERAD_START_MASK_H Register

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

Table 18-13. ERAD_START_MASK_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	START_MASK	R/W	0h	ERAD can generate upto 64 events per instance. This mask Is applied on the last 32 events. This mask if set generates a START DLT event, after which entries are recorded in the FIFO Reset type: SYSRSn

18.4.2.3 ERAD_END_MASK_L Register (Offset = 8h) [Reset = 00000000h]

ERAD_END_MASK_L is shown in Figure 18-5 and described in Table 18-14.

Return to the Summary Table.

ERAD End Mask for Lower 32 lines

Figure 18-5. ERAD_END_MASK_L Register

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

Table 18-14. ERAD_END_MASK_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	END_MASK	R/W	0h	ERAD can generate upto 64 events per instance. This mask Is applied on the first 32 events. This mask if set generates a END DLT event, after which entries are no longer recorded in the FIFO Reset type: SYSRSn

18.4.2.4 ERAD_END_MASK_H Register (Offset = Ch) [Reset = 00000000h]

ERAD_END_MASK_H is shown in Figure 18-6 and described in Table 18-15.

Return to the Summary Table.

ERAD End Mask for Higher 32 lines

Figure 18-6. ERAD_END_MASK_H Register

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

Table 18-15. ERAD_END_MASK_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	END_MASK	R/W	0h	ERAD can generate upto 64 events per instance. This mask Is applied on the last 32 events. This mask if set generates a END DLT event, after which entries are no longer recorded in the FIFO Reset type: SYSRSn

18.4.2.5 TAG_FILTER_START_REF Register (Offset = 10h) [Reset = 00000000h]

TAG_FILTER_START_REF is shown in Figure 18-7 and described in Table 18-16.

Return to the Summary Table.

Tag Filer Start Reference

Figure 18-7. TAG_FILTER_START_REF Register

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

Table 18-16. TAG_FILTER_START_REF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	START_REF	R/W	0h	TAG filtering: Recording in FIFO is started if incoming TAG when masked matches this reference Reset type: SYSRSn

18.4.2.6 TAG_FILTER_START_MASK Register (Offset = 14h) [Reset = 00000000h]

TAG_FILTER_START_MASK is shown in Figure 18-8 and described in Table 18-17.

Return to the Summary Table.

Tag Filer Start Mask

Figure 18-8. TAG_FILTER_START_MASK Register

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

Table 18-17. TAG_FILTER_START_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	START_MASK	R/W	0h	TAG filtering: Recording in FIFO is started if incoming TAG when masked with this MASK matches this reference Reset type: SYSRSn

18.4.2.7 TAG_FILTER_END_REF Register (Offset = 18h) [Reset = 00000000h]

TAG_FILTER_END_REF is shown in Figure 18-9 and described in Table 18-18.

Return to the Summary Table.

Tag Filer End Reference

Figure 18-9. TAG_FILTER_END_REF Register

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

Table 18-18. TAG_FILTER_END_REF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	END_REF	R/W	0h	TAG filtering: Recording in FIFO is stopped if incoming TAG when masked matches this reference Reset type: SYSRSn

18.4.2.8 TAG_FILTER_END_MASK Register (Offset = 1Ch) [Reset = 00000000h]

TAG_FILTER_END_MASK is shown in Figure 18-10 and described in Table 18-19.

Return to the Summary Table.

Tag Filer End Mask

Figure 18-10. TAG_FILTER_END_MASK Register

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

Table 18-19. TAG_FILTER_END_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	END_MASK	R/W	0h	TAG filtering: Recording in FIFO is stopped if incoming TAG when masked with this MASK matches this reference Reset type: SYSRSn

18.4.2.9 LINK_EN Register (Offset = 20h) [Reset = 00000000h]

LINK_EN is shown in Figure 18-11 and described in Table 18-20.

Return to the Summary Table.

Link Enable

Figure 18-11. LINK_EN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
LINK15_EN	LINK14_EN	LINK13_EN	LINK12_EN	LINK11_EN	LINK10_EN	LINK9_EN	LINK8_EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
LINK7_EN	LINK6_EN	LINK5_EN	LINK4_EN	LINK3_EN	LINK2_EN	LINK1_EN	LINK0_EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-20. LINK_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	LINK15_EN	R/W	0h	If bit is 1, corresponding LINK is allowed to record DLT entries in FIFO Reset type: SYSRSn
14	LINK14_EN	R/W	0h	If bit is 1, corresponding LINK is allowed to record DLT entries in FIFO Reset type: SYSRSn
13	LINK13_EN	R/W	0h	If bit is 1, corresponding LINK is allowed to record DLT entries in FIFO Reset type: SYSRSn
12	LINK12_EN	R/W	0h	If bit is 1, corresponding LINK is allowed to record DLT entries in FIFO Reset type: SYSRSn
11	LINK11_EN	R/W	0h	If bit is 1, corresponding LINK is allowed to record DLT entries in FIFO Reset type: SYSRSn
10	LINK10_EN	R/W	0h	If bit is 1, corresponding LINK is allowed to record DLT entries in FIFO Reset type: SYSRSn
9	LINK9_EN	R/W	0h	If bit is 1, corresponding LINK is allowed to record DLT entries in FIFO Reset type: SYSRSn
8	LINK8_EN	R/W	0h	If bit is 1, corresponding LINK is allowed to record DLT entries in FIFO Reset type: SYSRSn
7	LINK7_EN	R/W	0h	If bit is 1, corresponding LINK is allowed to record DLT entries in FIFO Reset type: SYSRSn

Table 18-20. LINK_EN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	LINK6_EN	R/W	0h	If bit is 1, corresponding LINK is allowed to record DLT entries in FIFO Reset type: SYSRSn
5	LINK5_EN	R/W	0h	If bit is 1, corresponding LINK is allowed to record DLT entries in FIFO Reset type: SYSRSn
4	LINK4_EN	R/W	0h	If bit is 1, corresponding LINK is allowed to record DLT entries in FIFO Reset type: SYSRSn
3	LINK3_EN	R/W	0h	If bit is 1, corresponding LINK is allowed to record DLT entries in FIFO Reset type: SYSRSn
2	LINK2_EN	R/W	0h	If bit is 1, corresponding LINK is allowed to record DLT entries in FIFO Reset type: SYSRSn
1	LINK1_EN	R/W	0h	If bit is 1, corresponding LINK is allowed to record DLT entries in FIFO Reset type: SYSRSn
0	LINK0_EN	R/W	0h	If bit is 1, corresponding LINK is allowed to record DLT entries in FIFO Reset type: SYSRSn

18.4.2.10 DLT_CONTROL Register (Offset = 24h) [Reset = 00000000h]

DLT_CONTROL is shown in Figure 18-12 and described in Table 18-21.

Return to the Summary Table.

DLT Control Register

Figure 18-12. DLT_CONTROL Register

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
KEY							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			FILTER_RST	CAP_MODE	TAG_FILTER_EN	ERAD_FILTER_EN	DLT_EN
R-0h			R-0/W1S-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-21. DLT_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	These 16 bits act as a key to enable writes to lower 16 bits of this register. The only time lower 16 bits can be updated is by a single 32-bit write where bits 31:16 equal 0x5a5a. All other writes are ignored including separate 16-bit writes. Read returns 0 for this field always. Reset type: SYSRSn
15-5	RESERVED	R	0h	Reserved
4	FILTER_RST	R-0/W1S	0h	1: Reset Filtering Unit, previous status derived from START and STOP will be cleared. Reset type: SYSRSn
3	CAP_MODE	R/W	0h	0: With each entry, TIME is captured 1: With each entry, PC source is captured Reset type: SYSRSn
2	TAG_FILTER_EN	R/W	0h	TAG based START-END filtering is enabled Reset type: SYSRSn
1	ERAD_FILTER_EN	R/W	0h	ERAD event based START-END filtering is enabled Reset type: SYSRSn
0	DLT_EN	R/W	0h	When 1, DLT Recording and triggers are enabled. When 0, logging is disabled Reset type: SYSRSn

18.4.2.11 FIFO_CONTROL Register (Offset = 28h) [Reset = 0003FF00h]

FIFO_CONTROL is shown in Figure 18-13 and described in Table 18-22.

Return to the Summary Table.

FIFO Control Register

Figure 18-13. FIFO_CONTROL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						WR_CTR_TRIG_LEVEL	
R-0h						R/W-3FFh	
15	14	13	12	11	10	9	8
WR_CTR_TRIG_LEVEL							
R/W-3FFh							
7	6	5	4	3	2	1	0
RESERVED						FIFO_RST	DMA_EN
R-0h						R-0/W1S-0h	R/W-0h

Table 18-22. FIFO_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17-8	WR_CTR_TRIG_LEVEL	R/W	3FFh	When WR_CTR_TRIG_LEVEL+1 new writes are received in FIFO after the last trigger, INT/DMA trigger is generated if enabled Reset type: SYSRSn
7-2	RESERVED	R	0h	Reserved
1	FIFO_RST	R-0/W1S	0h	Initialize write and read pointers (FIFO_PTR), number of words (FIFO_STS.WORD_CTR), Write counter of FIFO (FIFO_STS.WR_CTR) to 0 Reset type: SYSRSn
0	DMA_EN	R/W	0h	DMA request is generated with FIFO reaches FIFO_TRIG_LEVEL Reset type: SYSRSn

18.4.2.12 TIMER_CONTROL Register (Offset = 2Ch) [Reset = 00000000h]

TIMER_CONTROL is shown in Figure 18-14 and described in Table 18-23.

Return to the Summary Table.

Timer Control Register

Figure 18-14. TIMER_CONTROL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						TIMER2_RST	RESERVED
R-0h						R/W-0h	R-0h

Table 18-23. TIMER_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	TIMER2_RST	R/W	0h	TIMER2 is reset to 0 and stopped if this bit is set Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

18.4.2.13 FIFO_STS Register (Offset = 30h) [Reset = 00000000h]

FIFO_STS is shown in Figure 18-15 and described in Table 18-24.

Return to the Summary Table.

Number of entries in FIFO

Figure 18-15. FIFO_STS Register

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

Table 18-24. FIFO_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25-16	WR_CTR	R	0h	Write counter: Number of 64-bit words written to the FIFO after the last trigger This WR_CTR gets reset every time the WR_CTR = FIFO_TRIG_LEVEL Reset type: SYSRSn
15-10	RESERVED	R	0h	Reserved
9-0	WORD_CTR	R	0h	Number of 64-bit words to be read in the FIFO, i.e., current level of FIFO that is filled Reset type: SYSRSn

18.4.2.14 FIFO_PTR Register (Offset = 34h) [Reset = 00000000h]

FIFO_PTR is shown in Figure 18-16 and described in Table 18-25.

Return to the Summary Table.

Pointer locations in FIFO

Figure 18-16. FIFO_PTR Register

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

Table 18-25. FIFO_PTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25-16	WR_PTR	R	0h	Write pointer location of FIFO Reset type: SYSRSn
15-10	RESERVED	R	0h	Reserved
9-0	RD_PTR	R	0h	Read pointer location of FIFO Reset type: SYSRSn

18.4.2.15 TIMER2_COUNT Register (Offset = 38h) [Reset = 00000000h]

TIMER2_COUNT is shown in Figure 18-17 and described in Table 18-26.

Return to the Summary Table.

Timer2 Status

Figure 18-17. TIMER2_COUNT Register

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

Table 18-26. TIMER2_COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	Reserved
28-0	COUNT	R	0h	Current value of TIMER2 Reset type: SYSRSn

18.4.2.16 INT_FLG Register (Offset = 3Ch) [Reset = 00000000h]

INT_FLG is shown in Figure 18-18 and described in Table 18-27.

Return to the Summary Table.

Interrupt Flag

Figure 18-18. INT_FLG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		FIFO_TRIG	FIFO_UF	FIFO_OVF	TIMER2_OVF	TIMER1_OVF	INT
R-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 18-27. INT_FLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5	FIFO_TRIG	R	0h	FIFO reached Trigger level status Reset type: SYSRSn
4	FIFO_UF	R	0h	FIFO underflow status Reset type: SYSRSn
3	FIFO_OVF	R	0h	FIFO overflow status Reset type: SYSRSn
2	TIMER2_OVF	R	0h	TIMER2 overflow status Reset type: SYSRSn
1	TIMER1_OVF	R	0h	TIMER1 overflow status Reset type: SYSRSn
0	INT	R	0h	Global Interrupt Status Reading a 1 on this bit indicates that an interrupt was generated from one of the following events Reset type: SYSRSn

18.4.2.17 INT_EN Register (Offset = 40h) [Reset = 00000000h]

INT_EN is shown in Figure 18-19 and described in Table 18-28.

Return to the Summary Table.

Interrupt Enable

Figure 18-19. INT_EN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		FIFO_TRIG	FIFO_UF	FIFO_OVF	TIMER2_OVF	TIMER1_OVF	RESERVED
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 18-28. INT_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5	FIFO_TRIG	R/W	0h	FIFO reached Trigger level enable to generate an intterrupt Reset type: SYSRSn
4	FIFO_UF	R/W	0h	FIFO underflow enable to generate an intterrupt Reset type: SYSRSn
3	FIFO_OVF	R/W	0h	FIFO overflow enable to generate an intterrupt Reset type: SYSRSn
2	TIMER2_OVF	R/W	0h	TIMER2 overflow enable to generate an intterrupt Reset type: SYSRSn
1	TIMER1_OVF	R/W	0h	TIMER1 overflow enable to generate an intterrupt Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

18.4.2.18 INT_FRC Register (Offset = 44h) [Reset = 00000000h]

INT_FRC is shown in Figure 18-20 and described in Table 18-29.

Return to the Summary Table.

Interrupt Force

Figure 18-20. INT_FRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		FIFO_TRIG	FIFO_UF	FIFO_OVF	TIMER2_OVF	TIMER1_OVF	RESERVED
R-0h		R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h

Table 18-29. INT_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5	FIFO_TRIG	R-0/W1S	0h	FIFO reached Trigger level force Reset type: SYSRSn
4	FIFO_UF	R-0/W1S	0h	FIFO underflow force Reset type: SYSRSn
3	FIFO_OVF	R-0/W1S	0h	FIFO overflow force Reset type: SYSRSn
2	TIMER2_OVF	R-0/W1S	0h	TIMER2 overflow force Reset type: SYSRSn
1	TIMER1_OVF	R-0/W1S	0h	TIMER1 overflow force Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

18.4.2.19 INT_CLR Register (Offset = 48h) [Reset = 00000000h]

INT_CLR is shown in Figure 18-21 and described in Table 18-30.

Return to the Summary Table.

Interrupt Clear

Figure 18-21. INT_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		FIFO_TRIG	FIFO_UF	FIFO_OVF	TIMER2_OVF	TIMER1_OVF	INT
R-0h		R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 18-30. INT_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5	FIFO_TRIG	R-0/W1C	0h	FIFO reached Trigger level clear Reset type: SYSRSn
4	FIFO_UF	R-0/W1C	0h	FIFO underflow clear Reset type: SYSRSn
3	FIFO_OVF	R-0/W1C	0h	FIFO overflow clear Reset type: SYSRSn
2	TIMER2_OVF	R-0/W1C	0h	TIMER2 overflow clear Reset type: SYSRSn
1	TIMER1_OVF	R-0/W1C	0h	TIMER1 overflow clear Reset type: SYSRSn
0	INT	R-0/W1C	0h	Global Interrupt clear: Writing a 1 will clear the INT flag and enable further interrupts to be generated if any of the event flags are set to 1. Writing a 0 will have no effect. Reset type: SYSRSn

18.4.3 DLT_FIFO_REGS Registers

Table 18-31 lists the memory-mapped registers for the DLT_FIFO_REGS registers. All register offset addresses not listed in Table 18-31 should be considered as reserved locations and the register contents should not be modified.

Table 18-31. DLT_FIFO_REGS Registers

Offset	Acronym	Register Name	Protection
0h	FIFO_BUF_L	FIFO Content Register	
4h	FIFO_BUF_H	FIFO Content Register	
1000h + formula	FIFO_MEM_y	FIFO access in Memory mode	

Complex bit access types are encoded to fit into small table cells. Table 18-32 shows the codes that are used for access types in this section.

Table 18-32. DLT_FIFO_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

18.4.3.1 FIFO_BUF_L Register (Offset = 0h) [Reset = 00000000h]

FIFO_BUF_L is shown in Figure 18-22 and described in Table 18-33.

Return to the Summary Table.

FIFO Content Register

Figure 18-22. FIFO_BUF_L Register

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

Table 18-33. FIFO_BUF_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FIFO_BUF	R	0h	Contents of lower 32 bits of next FIFO location to be read Reset type: SYSRSn

18.4.3.2 FIFO_BUF_H Register (Offset = 4h) [Reset = 00000000h]

FIFO_BUF_H is shown in Figure 18-23 and described in Table 18-34.

Return to the Summary Table.

FIFO Content Register

Figure 18-23. FIFO_BUF_H Register

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

Table 18-34. FIFO_BUF_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FIFO_BUF	R	0h	Contents of higher 32 bits of next FIFO location to be read. A read to this register will increment the read pointer. Reset type: SYSRSn

18.4.3.3 FIFO_MEM_y Register (Offset = 1000h + formula) [Reset = 00000000h]

FIFO_MEM_y is shown in Figure 18-24 and described in Table 18-35.

Return to the Summary Table.

FIFO access in Memory mode

Offset = 1000h + (y * 4h); where y = 0h to 1FFh

Figure 18-24. FIFO_MEM_y Register

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

Table 18-35. FIFO_MEM_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FIFO_MEM	R	0h	Contents of FIFO location addressed Reset type: SYSRSn

Chapter 19

Waveform Analyzer Diagnostic (WADI)

This chapter discusses the features and functions of the waveform analyzer diagnostic (WADI).

19.1 WADI Overview

The waveform analyzer and diagnostic (WADI) peripheral consists of many useful built in signal analysis support and provides a safety mechanism for the signals. WADI is primarily useful for safety applications where driving switches or capturing signals require an action or a linking of actions to occur if the signal analysis reports any misbehavior.

19.1.1 Features

- Ability to select an input signal from multiple sources (CMPSS, ePWM, Input-XBAR, CLB, ADC) to WADI block and configure trigger to start analysis and perform safety diagnostics on the signals
- Ability to perform different checks as configured:
  - Pulse width measurement
  - Frequency measurement
  - Phase Overlap measurement
  - Dead-band measurement
- Ability to perform checks on individual signal or perform checks between two signals
- Ability to override outputs to a certain state or define a link of output combination based on analysis of signals
- Registers with parity support
- Support for RTDMA trigger and RTDMA acknowledgment

19.1.2 WADI Related Collateral

Foundational Materials

- C29x Academy - Waveform Analyzer Diagnostic (WADI)

19.1.3 Block Diagram

Figure 19-1 shows a block diagram of the WADI.

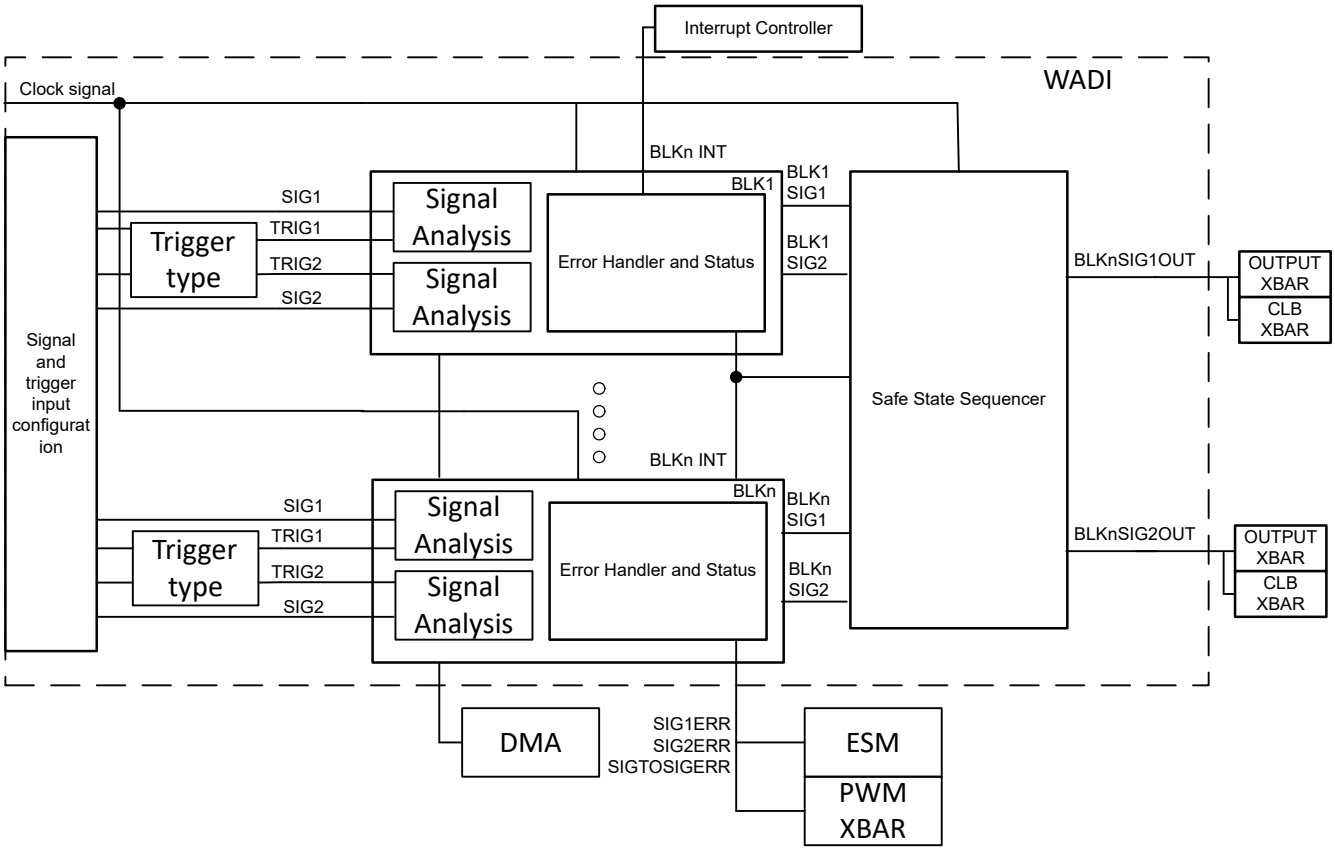


Figure 19-1. WADI Block Diagram

19.1.4 Description

The waveform analyzer diagnostic (WADI) determines the correctness and quality of the underlying real time control system by performing measurements, aggregation and comparison on the input signals. Each input signal is characterized for certain attributes of pulse width, frequency, phase, dead-band and so on. WADI validates the measurements for each signal against a compare value with some error of margin that the signal can still be considered valid. WADI allows comparison of individual signals or signal to signal analysis within a WADI block. There are four WADI blocks for each WADI instance. Each WADI block can monitor up to two signals and perform signal analysis on each.

19.2 Signal and Trigger Input Configuration

The WADI block is flexible in choosing signals to analyze. INPUTXBAR and PWMXBAR are routed to the WADI block (see Figure 19-2). There are two signals (SIG1, SIG2) that can be analyzed and two signals (TRIG1, TRIG2) that are used to start the capture of these signals. Each WADI block has two input signals and two trigger signals. Configuration for SIG1, SIG2, TRIG1, and TRIG2 are done through the BLKCFG register. TRIG1 triggers SIG1 analysis, and TRIG2 triggers SIG2 analysis.

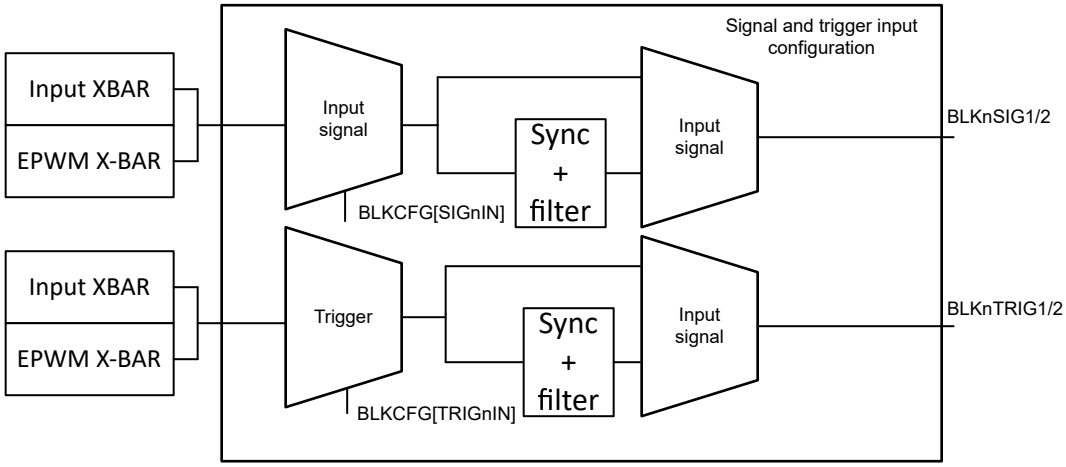


Figure 19-2. WADI Signal Input and Trigger Configuration

19.2.1 SIG1 and SIG2 Configuration

Cross-bar is used to provide flexibility to compare different input signals from various peripherals or external signals. Each WADI block can have access to INPUTXBAR and PWMXBAR for SIG1 and SIG2 respectively which can be configured in the BLKCFG[SIG1IN] and BLKCFG[SIG2IN] register.

SIG1 and SIG2 are the input signals for the WADI blocks. Table 19-1 shows the input sources for SIG1 and SIG2.

- SIG1 and SIG2 are configured independently of each other. Each WADI block has SIG1 and SIG2. For 4 WADI blocks, there are configurations for the 8 inputs (2 input signals per block).
- SIG1 and SIG2 can be configured to be synced and filtered before going to the signal analysis block. There is an added delay of up to 3-4 clock cycles when syncing and using the glitch filter.

Table 19-1. WADI Input Signals

WADI_SIG_I index	WADI1 Signal Name	WADI2 Signal Name
0	PWMXBAR1	PWMXBAR1
1	PWMXBAR2	PWMXBAR2
2	PWMXBAR3	PWMXBAR3
3	PWMXBAR4	PWMXBAR4
4	PWMXBAR5	PWMXBAR5
5	PWMXBAR6	PWMXBAR6
6	PWMXBAR7	PWMXBAR7
7	PWMXBAR8	PWMXBAR8
8	PWMXBAR9	PWMXBAR9
9	PWMXBAR10	PWMXBAR10
10	PWMXBAR11	PWMXBAR11
11	PWMXBAR12	PWMXBAR12
12	PWMXBAR13	PWMXBAR13

Table 19-1. WADI Input Signals (continued)

WADI_SIG_I index	WADI1 Signal Name	WADI2 Signal Name
13	PWMXBAR14	PWMXBAR14
14	PWMXBAR15	PWMXBAR15
15	PWMXBAR16	PWMXBAR16
16	INPUTXBAR33	INPUTXBAR49
17	INPUTXBAR34	INPUTXBAR50
18	INPUTXBAR35	INPUTXBAR51
19	INPUTXBAR36	INPUTXBAR52
20	INPUTXBAR37	INPUTXBAR53
21	INPUTXBAR38	INPUTXBAR54
22	INPUTXBAR39	INPUTXBAR55
23	INPUTXBAR40	INPUTXBAR56
24	INPUTXBAR41	INPUTXBAR57
25	INPUTXBAR42	INPUTXBAR58
26	INPUTXBAR43	INPUTXBAR59
27	INPUTXBAR44	INPUTXBAR60
28	INPUTXBAR45	INPUTXBAR61
29	INPUTXBAR46	INPUTXBAR62
30	INPUTXBAR47	INPUTXBAR63
31	INPUTXBAR48	INPUTXBAR64

19.2.2 Trigger 1 and Trigger 2

Trigger 1 and trigger 2 are used to apply the start condition for the signal analysis. The triggers have flexibility to use either of the two triggers for a specific WADI block. Table 19-2 shows the trigger selections for TRIG1 and TRIG2, respectively. This allows two entirely different sets of trigger inputs providing flexibility for triggering each SIGx. The trigger for SIG1 and SIG2 can be configured in BLKCFG[TRIG1IN] and BLKCFG[TRIG2IN] registers, respectively.

Table 19-2. WADI Trigger 1 Input Signals

WADI_TRIG_SIG1_I index	WADI1 Signal Name	WADI2 Signal Name
0	PWMXBAR9	PWMXBAR9
1	PWMXBAR10	PWMXBAR10
2	PWMXBAR11	PWMXBAR11
3	PWMXBAR12	PWMXBAR12
4	PWMXBAR13	PWMXBAR13
5	PWMXBAR14	PWMXBAR14
6	PWMXBAR15	PWMXBAR15
7	PWMXBAR16	PWMXBAR16
8	INPUTXBAR41	INPUTXBAR57
9	INPUTXBAR42	INPUTXBAR58
10	INPUTXBAR43	INPUTXBAR59
11	INPUTXBAR44	INPUTXBAR60
12	INPUTXBAR45	INPUTXBAR61
13	INPUTXBAR46	INPUTXBAR62
14	INPUTXBAR47	INPUTXBAR63
15	INPUTXBAR48	INPUTXBAR64

Table 19-3. WADI Trigger 2 Input Signals

WADI_TRIG_SIG2_I index	WADI1 Signal Name	WADI2 Signal Name
0	PWMXBAR1	PWMXBAR1
1	PWMXBAR2	PWMXBAR2
2	PWMXBAR3	PWMXBAR3
3	PWMXBAR4	PWMXBAR4
4	PWMXBAR5	PWMXBAR5
5	PWMXBAR6	PWMXBAR6
6	PWMXBAR7	PWMXBAR7
7	PWMXBAR8	PWMXBAR8
8	INPUTXBAR33	INPUTXBAR49
9	INPUTXBAR34	INPUTXBAR50
10	INPUTXBAR35	INPUTXBAR51
11	INPUTXBAR36	INPUTXBAR52
12	INPUTXBAR37	INPUTXBAR53
13	INPUTXBAR38	INPUTXBAR54
14	INPUTXBAR39	INPUTXBAR55
15	INPUTXBAR40	INPUTXBAR56

To start the analysis based on trigger, the configurations need to be complete before the WADI block is enabled.

The start of the signal analysis can be enabled for each of the signal measurement using triggers or if no triggers are configured the start of the first event detection right after the clock is enabled begins the signal analysis. SIG1 and SIG2 have individual configuration of the trigger which are no trigger, hardware trigger, software trigger, or synchronized trigger. Once the event triggers the measurements, the comparisons are continuous until the next trigger. If there is subsequent trigger then the measurement count restarts. If there are pending errors from previous counts, those are not reset, the application code must address the errors. Triggers selectable are categorized in Table 19-4.

Table 19-4. Trigger Types and Trigger Selection per WADI Block Signal

Selection	Trigger Type	Description
0x0	No Trigger (Default)	In this mode right after configuration for check is done on first event of configured type the measurement starts.
0x1	Hardware Trigger	This uses the hardware input from the WADI peripheral to start the measurement of signal.
0x2	Software Trigger	Each WADI block signal has an independent software bit control that the application can use to start the measurement.
0x3	Either Trigger	Whenever either hardware or software event is present the measurement is reset and started again.
0x4	Synchronized Trigger	This is special case when this is configured for signal then the triggers configured for other signal in WADI block is used for starting the measurement, this is useful for the signal to signal comparisons as well as having common trigger for related signals.
0x5-0x7	Reserved	All other configurations are reserved and those default to No trigger condition.

No Trigger condition is used when external trigger is not expected but continuous signal characterization is required. In this case once after WADI clock is enabled the event detection and measurement started. The WADI configurations must be configured before enabling clock. Similarly, if there is need to change the WADI block configuration then the clock must first be disabled followed by configuration change and enable of clock again. In case “No trigger” is set then clock enable acts as trigger, in all other cases detection and measurement waits for trigger.

On revision of settings, if there are stale measurements in WADI block then those are cleared and fresh measurements started for both SIG1 and SIG2. Rewriting the same setting again does not change the status or ongoing waveform measurement.

19.3 WADI Block

19.3.1 Overview

WADI block has multiple modes and types of checks to analyze the input signals. There are dedicated counters to aggregate the number of measurements, counters for pulse width, edges, dead-band and phase overlap. To verify the signals there are comparisons and measurement error events. For cases where simultaneous modes are needed but the modes are exclusive then another WADI block must be used to analyze the signals. For example, pulse width measurement and edge count analysis cannot be used together in the same block. Below descriptions give overview of some of the common hardware components of WADI block

19.3.2 Counters

Each input signal into the WADI Block (SIG1 and SIG2) have independent counters to perform signal analysis for either pulse width measurement or edge count measurement along with signal to signal analysis.

1. Width of the time between two configured edges.
2. Number of signal edges within the fixed time interval to determine the signal frequency.

The counter can start based on the trigger configured in the trigger settings. If there are no triggers configured, and the WADI block is still enabled, the counter begins upon first programmed edge of SIGx.

The counter restarts for next iteration once the pulse width or frequency measurement is completed. Triggers can be programmed such that SIGx within WADI block or multiple signals across WADI blocks can be synchronized together to start the operation using hardware or software triggers. Thus all the SIGx counters in WADI IP can be used as standalone counters and measurements or be synchronized to common event.

In case there is assertion of configured trigger while the measurement is on-going then the respective block restarts the measurement and aggregation. Any errors which are detected in previous cycles are not cleared and those need to be explicitly cleared by the software.

Note

Pulse width and Frequency are mutually exclusive modes hence common counter is used to measure either based on the configuration of width or edge count and corresponding edge/time window details.

Counter for aggregating number of measurements

Each of the SIGx of WADI block has capability to accumulate pulse width and frequency readings. This accumulation is used in multiple ways :

- For Pulse width accumulation the aggregation depicts the total assertion of signal with particular polarity configured within the SIGxCFG[SIGPOL] register. Such a sum is indicative of total energy transferred over number of accumulated readings. The number of readings to accumulate are configurable in the SIGxCFG[NUMAGGR] register. Same accumulator used for frequency reading accumulation.
- This accumulator output and margin can be directly compared to a compare value defined in the register SIGxCMPA/B, SIGxPKCFG, and SIGxAVGCFG. Comparison also provides flexible margin(+/-) to allow for variation between two different types of systems.
- Secondly the average of the accumulated value at the end of the accumulation can also be compared. Given the power of 2 units accumulation the average is based on right shift of the accumulator value in hardware (no complex division is expected in hardware). An independent counter that tracks the number of measurements done on each signal.

Counter for dead band and phase overlap

In addition to individual signal analysis, dead band and phase overlap between two signals can also be done. The WADI block handles the logic to compute the signal to signal differences to perform signal to signal analysis. The dead band and phase overlap checks directly count the time difference between edges of two input waveforms on WADI block, hence separate counter is used for each of these modes. Both of these modes are mutually exclusive and based on check selected.

1. Dead-band: Based on edge type programmed, the distance between SIG1 to SIG2 edge is measured. Only EDGE_TYPE is applicable for this check, EDGE_SPAN is not. Whether to check configured EDGE_TYPE pairs from SIG1 and SIG2 or also check the opposite pairs is configured by DBAND_CHK_TYPE.
2. Phase Overlap: Uses the configuration of EDGE_TYPE similar to Dead-band. The end state of the configured edge is considered as level under review. For example, if SIG1 is configured as rise edge and SIG2 is configured with fall edge then the high “level” assertion of SIG1 simultaneous with low “level” assertion is counted. Normal expectation is that signal that asserted first can also be de-asserted prior to other signal de-assertion, but hardware only considers simultaneous configured level assertions of both signals.

19.3.3 Pulse Width

There are 3 types of measurements associated with pulse width:

1. Pulse width single measurement
2. Pulse width aggregation
3. Pulse width average and peak

19.3.3.1 Pulse Width Single Measurement

- Single pulse width measurement means to validate a single waveform for time width between two programmed edges. The time between two edges of given waveform is measured in terms of input clock to the block. Measurement is based on the configuration of the edge type and edge numbers.
- The pulse width single measurement and pulse width aggregation are mutually exclusive, one can do individual edge-to-edge width check or does sum check. The software configured values come from the same compare registers; hence, only either the individual readings or accumulated readings are compared.
- The WADI block measures the pulse width of the signal waveform. On which edge the counters start is defined in the SIGxCFG[SIG_POL] that can be on rising, falling, or either edge. The SIGxCFG[EDGESPAN] defines how many edges to span to measure the width.
- Figure 19-3 shows a few examples of how to measure a given pulse width. For example, if SIGxCFG[EDGESPAN] = 0x1, the pulse width being measured is from 1 to 2. If SIGx_CONFIG[EDGE_SPAN] = 0x2, the pulse width being measured is from 1 to 3.
- Configuration to span across more than the third edge is possible and provides flexibility for multiphase or multilevel waveforms.

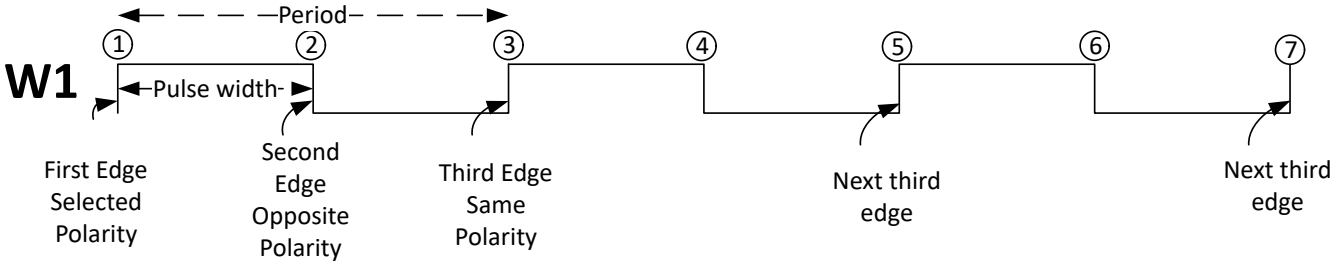


Figure 19-3. Pulse Width Diagram

19.3.3.2 Pulse Width Aggregation

Pulse width aggregation is the accumulated width measurement for configured number of units. For example, pulse width aggregation for every 2nd edge for 32 times. Pulse width aggregation readings are distinct and after one accumulation cycle is over the next one is started without any history of accumulation. However if there is an error in the previous cycle the aggregation results stay until software clears the error.

- SIGxCFG[NUMAGGR] is the register to configure the number of measurements configurable only in terms of power of 2 – 2,4,8,16,32,64.
- If aggregation is enabled, single measurement pulse width can aggregate instead of being independently compared.

19.3.3.3 Pulse Width Average and Peak

WADI block signal analysis can determine both the peak (highest value) and the averaging of the pulse width aggregation (Average of units as power of 2 hence simpler division)

- SIGxCFG[AGGRMODE] register selects the type of check to be enabled whether only aggregate(sum), peak or average or combinations of the same.

19.3.4 Edge Count

To characterize the frequency of a single waveform, WADI counts the number of edges in the prefixed time window set by the SIGxEDGECFG register. To use the edge counter, the SIGxEDGECFG[CNTEN] bit must be set.

- The time window within which the edges are counted is defined with the SIGxEDGECFG[TIMEWNDOW]. The window size is in terms of SYSCLK in which the edges are counted. The time window starts upon event of the configured trigger (start of edge counting) and ends upon lapse of configured window length (edge counting is stopped). The count begins on the edge defined by SIGxCFG[SIGPOL].
- The SIGxCFG[SIGPOL] is useful to start counting specific edge. While SIGxCFG[EDGE_SPAN] allows skipping intermediate edges. For frequency or edge count measurement, the recommended configuration for SIGxCFG[EDGE_SPAN] is 0x2. This captures the second edge from the previous edge detected, which counts the same edge type (that is, rise edge to next rise edge) and measures the frequency of the waveform in the time window. If the SIGxCFG[EDGE_SPAN] is configured greater than 2, then edges counted accordingly skipping intermediate edges.

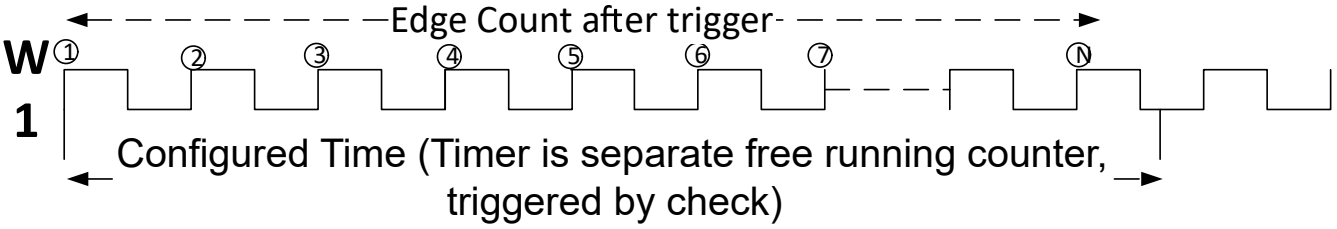


Figure 19-4. Edge Count Diagram

19.3.4.1 Edge Count with Fixed Window

Frequency uses the SIGxCMPA values, margin, edge type to count, average, peak, comparison settings.

The register to configure a fixed time window to analyze the edge counts is defined in SIGxEDGECFG[TIMEWNDOW].

19.3.4.2 Edge Count with Moving Window

This comparison counts edges in moving time window. Moving time window is the same time window but with overlap start within existing window. This allows better granularity to check that edge counts are uniform in the window and if those are changing the transition is smooth.

For example:

1. If the moving timing window (MVWTIME) is configured to 2000 time-base counts and number of readings (MVWCNT) is configured to be 3 (0x3).
2. Then after trigger signal edges are counted for every 2000 time-base count. Readings of multiple (2 or 3), in this case (MVW_CNT) 3, moving windows are accumulated to make one fixed window.
3. Thus the size of the fixed window is 6000. After first fixed window measurement is over, the measurement is compared against SIGxCMPA/B and the corresponding margin to verify the measurement is within the minimum-maximum range.
4. Further upon next moving window reading accumulation, it is added to fixed window measurement and the earliest reading is retired (subtracted) from the signal edge count of the fixed window. Thus the reading to be compared is always maintained for one fixed window. For example, Windows intervals successively progress for time base counts of 0-6000, 2000-8000, 4000-10000, 6000-12000, so on.

- The given size of the moving window is configured in the SIGxEDGEMVWCFG[MVWTIME] and the number of moving windows SIGxEDGEMVWCFG[MVWCNT] is to be accumulated.
- Moving window count MVWCNT is limited to 1, 2, 3, or 4 of which a count of 1 means that there is no moving window. Either bisect or trisect the fixed window with as many readings (3 readings for bisect and 4 for trisect) to manipulate.
- As depicted multiple accumulator snapshots can be taken for the measurement count at the interval of MVWTIME and readings are computed to find the number of edges within MVWTIME * MVWCNT period. Check of readings after every snapshot gives granular movement of window.

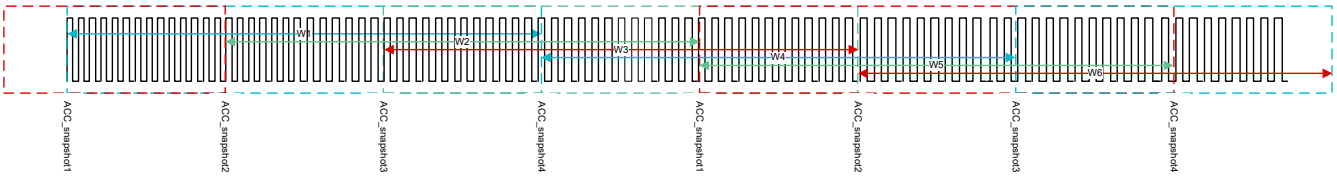


Figure 19-5. Edge Counting of Signals for Moving Window

19.3.5 Signal1 to Signal2 Comparison

To use signal to signal comparison for pulse width, edge count, phase, deadband the BLKTRIGCFG register must be configured such that SIG1 and SIG2 have a common trigger input. This can be done by configuring the BLKTRIGCFG[SIGx_TRIG_TYPE] to set either SIG1 or SIG2 to be set to synchronized trigger.

Pulse Width SIG1 to SIG2 Comparison

- Signal to signal comparison is enabled by the SIGTOSIGCFG[SIGTOSIG_CMPEN] register. The common configuration to start signal comparison is to have a hardware trigger on one SIGx, and configure a synchronized trigger for the other signal. This starts counting from a common trigger point for both signals. This trigger mechanism of synchronized use is common for all the SIGTOSIG operations including width, edge-count, dead-band, and phase overlap.
- In case the SIGTOSIG compare enabled and synchronized trigger is not configured to either SIG, then hardware forces SIG1 trigger to be used for both SIG1 and SIG2.
- A valid signal to signal comparison is when there are pulses occurring on both SIG1 and SIG2. An example of an invalid signal to signal comparison is when comparing the widths of two signals and there is no pulse on the second signal. If the next pulse is started on the first signal, and second signal does not have a pulse then the earlier reading of the signal is discarded and no comparison is reported.
- In such case, the error reported within the BLKERRSTS[SIGTOSIG_ERR] register can be raised to indicate that after trigger there was imbalance in compare points of two waveforms. The BLKERRINFO[ERRCNT] register reflects the measurement count and the type of error to log in this register is defined in the BLKERRINFO[ERRTYPE] register. Like any other event that is triggering interrupt or RTDMA request, BLKERRSTS[SIGTOSIG_ERR] also causes an interrupt or RTDMA request.
- The expected compare value and margin for the difference between width of two signals can be defined in the SIGTOSIGCFG[CMP] and SIGTOSIGCFG[CMPMARGIN] registers.

Pulse width Aggregation signal to signal comparison

- Signal to signal aggregation comparison: This comparison is exclusive of single pulse width measurement and compares the difference of aggregated measurements of two signal values. The width of signals is measured as per edge configuration and such N successive measurements accumulated. Where N is the configured aggregation count. The difference is taken when both readings are available and comparison is performed. Compare values are programmed by user through SIGTOSIGCFG[CMP] and SIGTOSIGCFG[CMPMARGIN]. These are same registers used for single pulse width comparisons.

Pulse width average, Pulse width peak signal to signal comparison

- Pulse width peak comparison is detected for both signals and then the difference of peak values is compared to the programmed threshold with margin.
- Pulse width average signal to signal comparison is taken the average of accumulated value of N measurements of both signals and finding the difference of average compared to the programmed threshold with margin.
- Both peak and average have separate compare values available hence these both checks can be simultaneous and can also be concurrent with each other.

Edge count comparison

Edge count signal to signal comparison can be done in two ways:

1. Fixed window comparison
2. Moving window comparison

While comparing the edge count, in a given interval if there are zero edges from another signal, then the compare is not a valid comparison point.

- Comparing edge counts of two signals on moving window basis. This compares the fixed window accumulation for every moving window completion between SIG1 and SIG2. Given that clock is common for WADI block and same between SIG1 and SIG2, the same settings of moving time window are used for both SIG1 and SIG2. This needs to be made sure by software. Any mismatch leads to erroneous results that do not help in comparison. Note the check can be simultaneous with MVW mode of individual signal characterization; hence, no hardware restrictions are imposed.

Comparing the number of edges between each interval such that there are no glitches or excursion within small windows and at the same time the check of time-window having certain frequency range is covered. The check works in similar manner as basic window comparison except that for basic moving window check the differences between the counts is compared against SIG_TO_SIG_CONFIG[DIFF_CMPARE_CNTRE] and with SIG_TO_SIG_CONFIG[DIFF_CMPARE_MARGIN] margin.

Note

Pulse width and pulse width aggregation versus frequency fixed time window and frequency with moving window are mutually exclusive. The WADI block for accumulating measurement, and calculating the difference between measurements must be for either pulse width or frequency measurement.

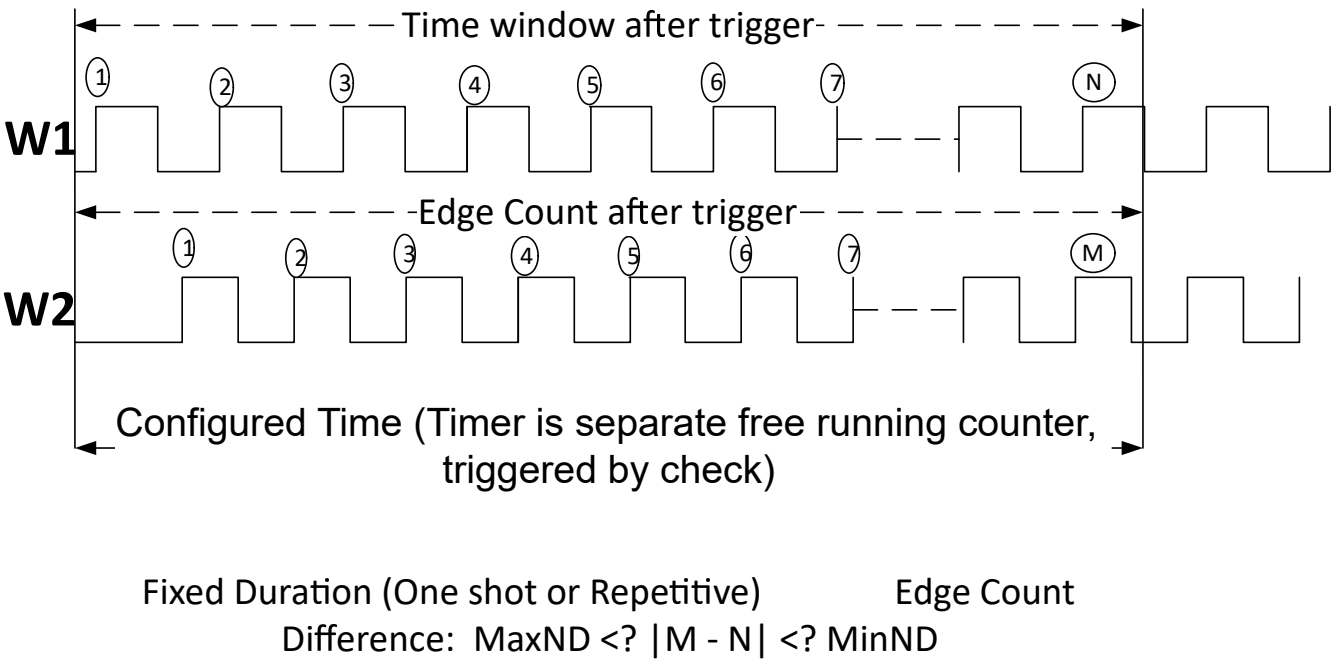


Figure 19-6. Signal to Signal Edge Count Comparison

19.3.6 Dead Band and Phase

Dead Band

Dead band and phase overlap checks are based on specific edges hence whether specific edges as configured in SIGPOL or ones opposite to the SIG or both to be compared is critical to the configuration.

Note

EDGESPAN configuration is not used for dead-band check as edges of two signals of the configured type are compared on continuous basis for any violation on the minimum width. Check starts after trigger to search for configured edge type for each signal.

This configuration of singular or both edges of dead band check is done in SIGTOSIGCFG[SIGTOSIG_DBCMPEN]. The order of edges can be either SIG1 or SIG2, rise or fall but only edges which are configured need to be measured. Dead band looks only for distance between edges if less than compare value defined in SIGTOSIG_DBOLAPA/B[DBCMPA/B].

- When SIGTOSIGCFG[SIGTOSIG_DBCMPEN] bit is enabled and SIGTOSIGCFG[DBCMPTYPE] = 0x0, this configuration compares the distance between configured edges of the respective signal. For example, if SIG1-rise and SIG2-fall is to be compared, then after trigger on SIG1-rise occurs then distance to SIG2-fall can be compared to the compare value.
- When SIGTOSIGCFG[DBCMPTYPE] = 0x1 with SIGTOSIGCFG[SIGTOSIG_DBCMPEN] enabled, this configuration allows either comparing configured or the compared and complementary edges for dead-band. Complementary check can be specifically enabled. For example, if configured edges are SIG1-fall and SIG2-rise, then upon enabling both SIG1-fall to SIG2-rise and SIG1-rise to SIG2-fall is checked.
- The edge to edge distance is measured as per configuration continuously edge after edge. If SIG1 edge of programmed type (for example, Rise) is received then counter starts counting till SIG2 edge of configured type (for example, Fall) occurs.

Note

For the dead-band check the EDGE_SPAN configuration is ignored and the comparison is performed for every two associated or associated + complementary edges.

1. Same is applicable if the order of signals reverses and SIG2 (for example, Fall) occurs ahead of SIG1 (for example, Rise).
2. In case dead-band check is enabled for either edge (DBAND_CHK_TYPE=0x1) then upon detecting edge on either signal say SIG1 (for example, Rise) in this case, corresponding configured edge on SIG2 (for example, Fall) is detected and the interim period is checked for meeting at least required threshold compare.
3. Further for case in 2. The opposite edge of either signal configured in EDGE_TYPE (for example, SIG2.Rise) is detected and the distance to corresponding opposite edge of other signal (for example, SIG1.Fall) is validated. Thus both edges of signals are validated against the corresponding edges of paired signal.
4. In case after detection of specific edge (for example, SIG1.Rise), if corresponding edge (for example, SIG2.Fall) for paired signal does not occur before the opposite edge (for example, SIG1.Fall) of the same signal then the count is reset and check is started for the recent edge.

Phase Overlap

Phase measurement also takes settings similar to dead-band, except that the level after configured edge is measured for overlap. Order of signal occurrence is not concern as overlap is measured.

- Signal to Signal phase overlap : Checks particular signal levels (say SIG1-high and SIG2-low) are simultaneously asserted with time overlap within range of minimum to maximum. Measurement starts when both signals at configured level. Whereas upon exit of either signal from the level sought, the measurement is stopped and compared against SIGTOSIG_DBOLAPA/B[DBCMPA/B] with margin of +/- OLAPCMPA/BMARGIN
- In this mode of operation, expected the waveforms provided for comparison do not have individual pulse widths such as one signal has pulse more than 2^x wider than other. If for some reason while one of the signals remains asserted and the second signal gets reasserted, then a second time the measurement is triggered flushing the previous measurement. However, if there are errors raised in earlier comparisons then those are not deasserted until cleared by software.
- SIGTOSIGCFG[OLAPCMPTYPE]: Enables and selects the type of overlap comparison. The following table indicates the configuration and the effect on the checks. The example is for the configuration of rise edge of the SIG1 and Fall edge of the SIG2.
- For the phase-overlap check the EDGESPAN configuration is ignored and the comparison is performed for every two configured or configured complementary edges.

Control	Value	Check	Check for Example
OLAPCMPTYPE	0x0	0x0 - Overlap of levels per configuration (edge configuration extended to derive level).	Overlap of SIG1-level-High and SIG2-level-Low
OLAPCMPTYPE	0x1	0x1 - Overlap of levels per configuration and complementary levels as well.	Overlap of SIG1-level-High and SIG2-level-Low and Overlap of SIG1-level-Low and SIG2-level-High

19.3.7 Simultaneous Measurement

WADI block can measure the pulse width, frequency, dead band, and phase overlap. WADI block can be configured for single measurements or be configured to instead calculate an aggregated measurement and take the average or find the peak value of the measurements.

Each WADI block can be configured to measure either pulse width of a signal or the frequency (edge counts) of a signal. Configuration for measuring both pulse width and frequency on one signal cannot be done. However configuration to measure frequency on one signal, and pulse width on another signal is allowed.

Single signal measurements can be done with out any signal to signal comparison. There is flexibility to perform signal to signal measurements on top of single signal measurements.

For example, you can measure the pulse width in addition with a signal to signal measurement of dead band and phase overlap. The following scenarios show the possible simultaneous measurements that can be done depending on what is needed to be measured.

For simultaneous measurements, here are the three configuration scenarios that are feasible with each WADI block. Dead band and phase overlap measurements are possible in all scenarios since an internal independent counter is used for them. The following scenarios go through what can be measured on SIG1 and SIG2 independently, and what can be done when performing a SIG1 to SIG2 comparison.

Scenario 1 (Pulse width and frequency measurement):

One WADI block can be used to measure SIG1's pulse width and measure SIG2's frequency independently. In this scenario you cannot use any of the signal to signal comparisons since one is measuring pulse width and another measuring frequency and the measurements are not be related to each other. You can measure the individual signals for the single measurement or if you enable measuring the signal's sum, average, and peak.

Scenario 2 (Pulse width only):

One WADI block can be used to measure pulse width of SIG1 and SIG2. During the scenario, you can measure the independent pulse widths and do signal to signal comparison. Signal to signal comparison can be done since both counters have pulse width measurements related to each other. If the WADI block is measuring a single pulse width (SIG1/2CFG.NUMAGGR = 0x0) then sum, average, and peak comparisons as well the signal to signal for sum, average and peak can not be used for comparison. If SIG1/2CFG.NUMAGGR is non-zero, then the sum, average, and peak comparisons can be used and can be enabled using SIG1/2CFG.AGGRMODE. In this scenario frequency comparison can not be available since we are measuring pulse width.

Scenario 3 (Frequency only):

To enable measurement of frequency or edge count, SIG1/2EDGECFG.CNTEN must be set. One WADI block can be used to measure frequency or edge count of SIG1 and SIG2. During the scenario, you can measure the edge count, or measure a moving window of edge counts. In addition you can do signal to signal comparison only for comparing the single edge count measurement or the moving window measurement. Signal to signal comparison can be done since both counters have edge count measurements related to each other. Sum, average, and peak comparisons as well the signal to signal for sum, average and peak can not be used when in edge count mode for comparison. In this scenario pulse width comparison or pulse width measurement can not be available since we are measuring edge count.

19.4 Safe State Sequencer (SSS)

- WADI can analyze the input waveforms and has the flexibility to override the input waveform to a desired event of outputs required to meet safety or efficiency requirements. The corrections to input waveform can be set to a fixed state after certain events or chain the output of events in a specific manner or do nothing.

The outputs of the WADI blocks are routed to the Safe State Sequencer(SSS). The SSS provides a safe state handling method for processing the signal upon a fault or error during signal analysis. Once an error is detected or a pattern of error events are detected, the SSS can generate a define pattern of outputs.

The SSS has a number of event triggers and safe state event outputs that is used to drive the outputs to a known state. In order to enable SSS, a minimum of 4 events and 4 event outputs are needed. The maximum are 8 events and 8 event outputs can be used. Key functions/blocks and their relation is depicted in figure below.

19.4.1 SSS Configuration

The WADI's signal analysis block (see Figure 19-7) generates error events related to SIG1 and SIG2 that can be used to start the SSS block. The SSS block essentially provides a waveform pattern generation based off the WADI block signal analysis events. The SSS has custom configurations for how long to hold a signal low or high. The trigger or multiple trigger configuration block is how the SSS is enabled. The output event + output event trigger + duration + link configuration block are flexible settings that can be used to control the outputs, control which events triggers the outputs, control how long to hold the outputs in a high or low state, and allows cyclic signal generation. For example, if an error event occurs from the WADI block's signal analysis the SSS can output a pattern of high and low signals in a cyclic manner or perform the pattern of outputs once.

Each WADI's block signal analysis events can be used to start the SSS block. There are a total of eight configurable events (SSS_EVTnCFG[EVTn]) that can be used to compare against the trigger event (SSS_EVTTRIG) for SSS to start. Each configurable event holds the settings for all WADI blocks 1 to n. The trigger event triggers the start of the SSS if the trigger event matches one of the configured SSS events (SSS_EVTnCFG[EVTn]). Once the trigger event is met with the SSS event, the signal SIGn is overwritten with the configurations defined in the SSS output event configuration (SSS_BLKSOUTEVTnCFG[OUTEVTn]). The

SSS output event can either set the output high or low. After defining the output event configurations, the SSS maps the output event to the WADI block's signal output.

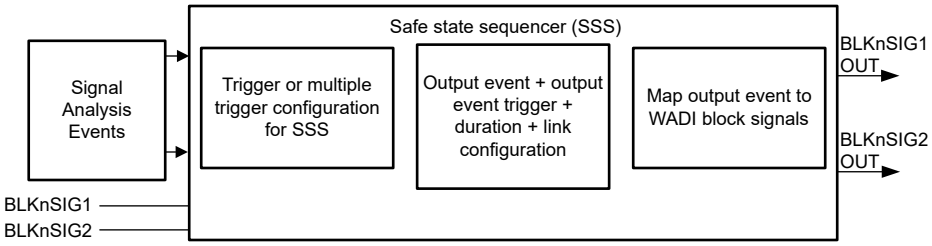


Figure 19-7. SSS High-Level Block Diagram

Steps for output event configuration (see Figure 19-8) for WADI block 1 and 2 example:

1. Configure what action to occur for output event 1-8 using SSS_BLKSOUTEVTnCFG[OUTEVTn]
2. Configure which output events can control which block signals using SSS_BLK1_2OUTSEL[BLK1SIG1], SSS_BLK1_2OUTSEL[BLK1SIG2], SSS_BLK1_2OUTSEL[BLK2SIG1], SSS_BLK1_2OUTSEL[BLK2SIG2]

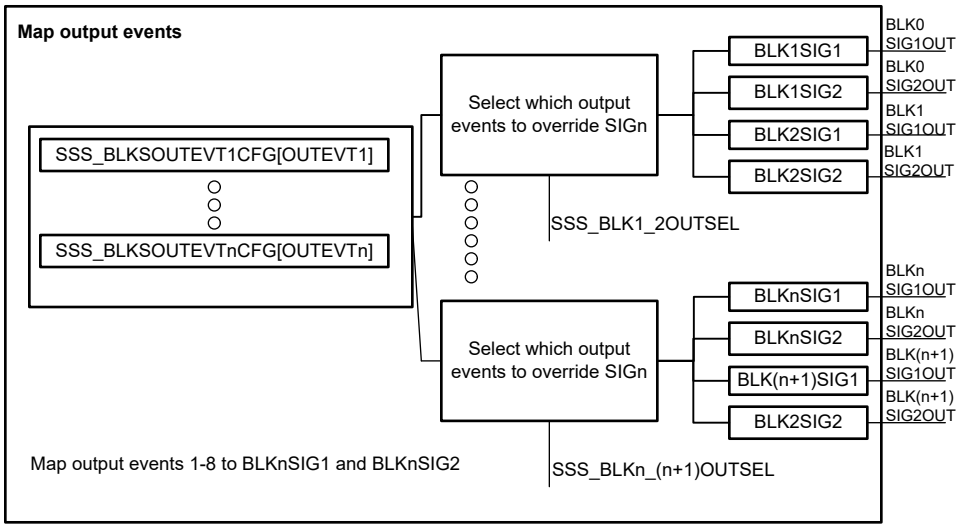


Figure 19-8. Map Output Events to BLKnSIGn

There are four common ways to configure the SSS.

1. Single trigger with single pattern output events: Used for single trigger based enabling of SSS
2. Single trigger with cyclic pattern output events
3. Multiple trigger with single pattern output events: Used for chaining event triggers before enabling SSS
4. Multiple trigger with cyclic pattern output events

Figure 19-9 is a general diagram of configuring the SSS for single trigger with a single pattern of output events. Each output event can be configured to trigger on different events. For example, OUTEVT1 can be triggered on EVT1, EVT2, EVT3, and so on, using SSS_OUTEVTnTRIGCFG. To configure the duration of the output event, that is done through the SSS_OUTEVTnDUR.

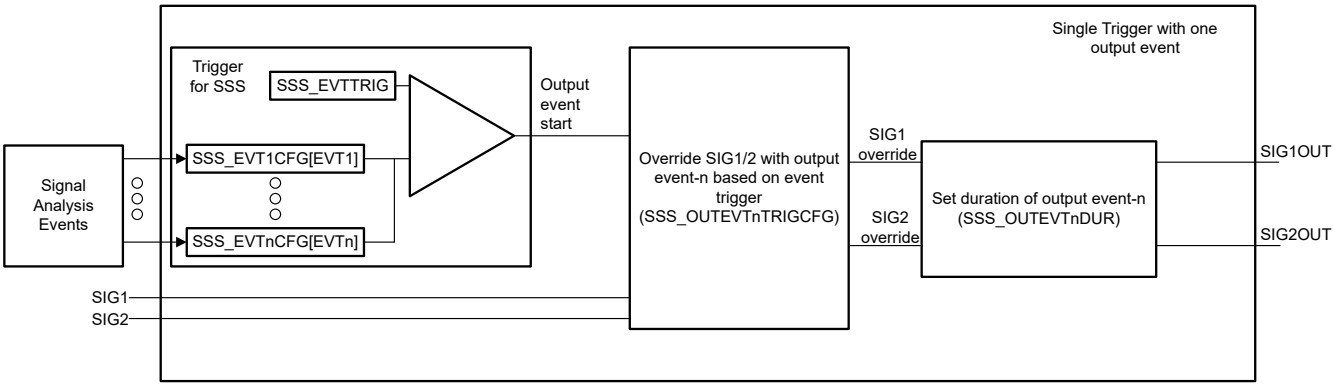


Figure 19-9. Single Trigger with Single Pattern Output Event

Each output event has a fixed time of activation unless the output event is configured to be always high or low. This time is configured using the SSS_OUTEVTnDUR count that is configured by user and starts down-counting once trigger to activate the output event is met. Once the counter reaches 0, the output event is removed and delay word is reloaded for next activation. User can configure the count to 0xFF_FFFF, which is the maximum count and that is used to keep the output event continuously active. This is a special setting used for driving fixed value safe state output.

In case the subsequent sequence word is configured to be linked and shares the outputs, then make sure that before configuring the fixed overrides to 0xFF_FFFF (to delay count), the configuration either changes the output sharing or link condition of subsequent sequence word to avoid an unpredictable condition.

By hardware design, the delay condition overrides and the subsequent sequence word steps are ignored if delay count is configured to be 0xFF_FFFF.

If the second trigger to occur while sequence word is active then count is re-initialized to configured count and down-count is restarted again. Thus extending the sequence word application to related outputs.

Sequence delay setting is an independent word for each sequence word, SSS_SEQUENCE_DELAYx. The setting is a 24-bit setting that allows a delay of up to 40ms (400MHz input clock - without division) that is sufficient to drive the intermediate state of switch activations in present topologies.

Figure 19-10 shows multiple trigger with single pattern output events.

For configuring multiple triggers, in Figure 19-10, EVT1 must occur before moving onto the second event trigger. Once that event has occurred, then EVT2 must happen for the SSS to start.

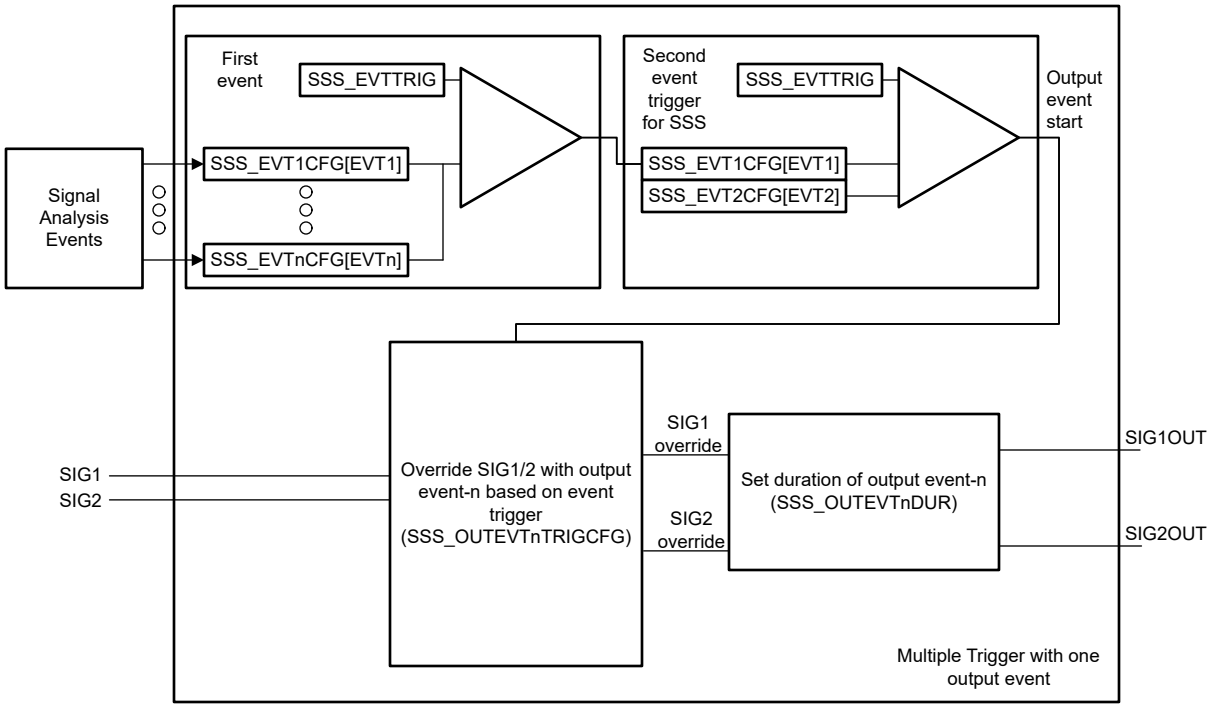


Figure 19-10. Multiple Trigger with Single Pattern Output Events

Figure 19-11 shows a multiple trigger configuration before the start of the SSS and linking output events to each other to generate a single pattern of output events. For example, if the desired output is four output events high, low, high, low, the SSS can do this with linking output events to one another. The linking can be done through the SSS_OUTPUTEVTnLINKCFG.

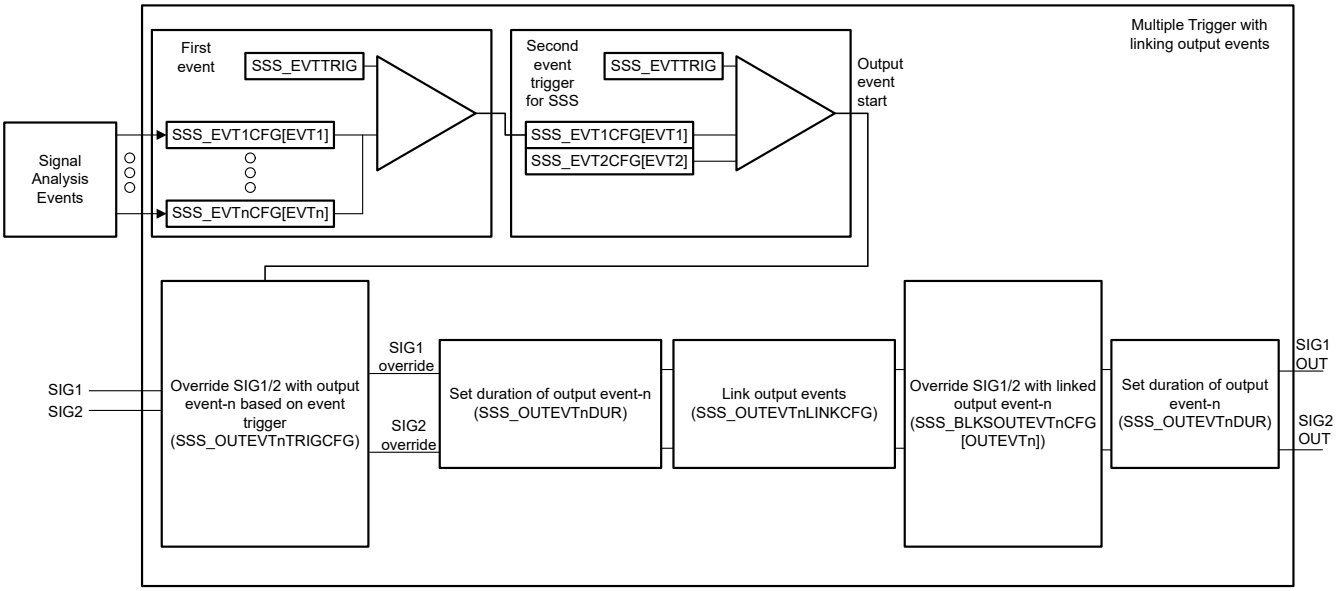


Figure 19-11. Linking Output Events for Generating Pattern Output Events

To generate a cyclic pattern of output events, Figure 19-12, you can do so by linking the output events in a circular fashion. For example if the waveform generation pattern requires a continuous cyclic pattern of high, low, high, low. The output events 1, 2, 3, 4 can be used. Then linking output events 1 → 2 , 2 → 3, 3 → 4, then configure output event 1's link to be output event's 4. This creates the cycle of the pattern generation to continuously output this linking of output events.

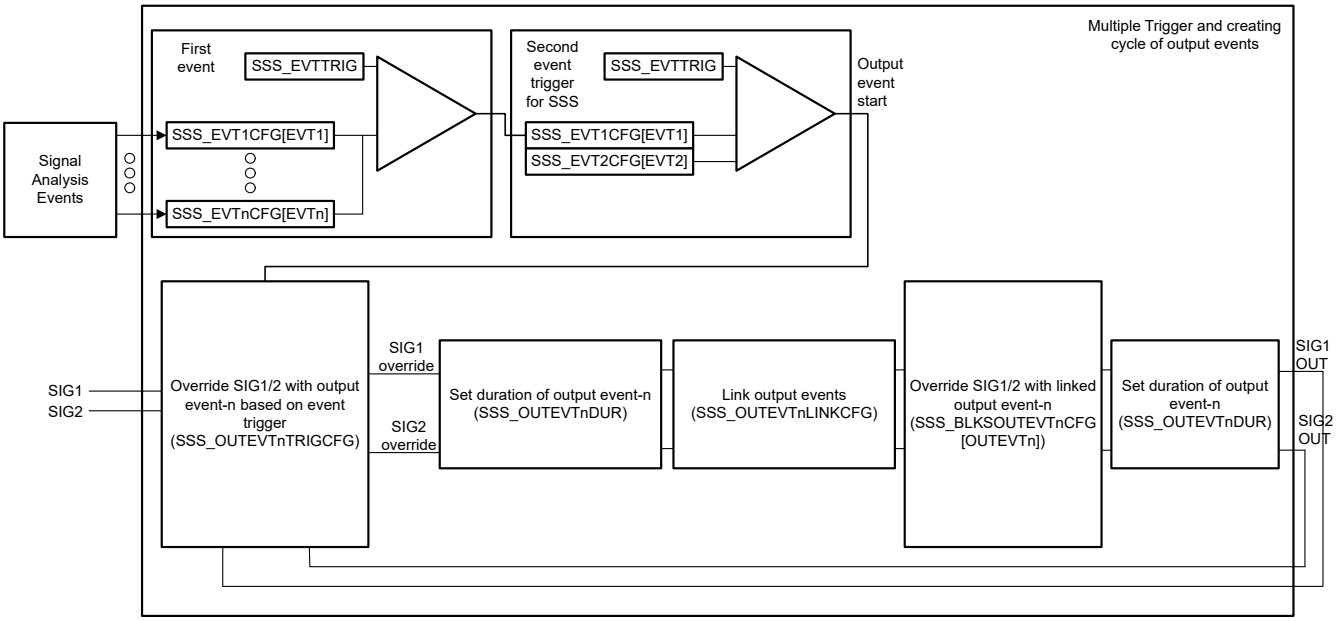


Figure 19-12. Multiple Trigger with Cyclic Pattern Output Events

Configuration of multiple triggers

In intricate switch topologies, a single event can not reflect the full condition to trigger the SSS. The topology can need two or more states to fully detect the failure condition before starting the safe state sequences. In such cases, the SSS allows linking multiple trigger events together. The register to configure multiple triggers is SSS_TRIGEVT1_4CFG and SSS_TRIGEVT5_8CFG.

Note

To configure SSS_TRIGEVT1_4CFG or SSS_TRIGEVT5_8CFG, the user must write the compliment before the actual value. For example, if four trigger events are needed before SSS is enabled writing 0x1999 does not work. Instead writing 0xE1_69_69_69 is required to configure SSS_TRIGEVT1_4CFG.

Table 19-5. Multiple Trigger Configuration

Event Word Select Combination				Words Combination for Sequences	Trigger Numbering	Action for Event Word Compare
Event 4	Event 3	Event 2	Event 1		Trig# (EVT#, next EVT#,...)	
0	0	0	0	0	NA	None of the event words are used and safe-sequencer is not used.
1	1	1	1	1+1+1+1	T4(EVT4), T3(EVT3), T2(EVT2), T1(EVT1)	Each event trigger is independently configured and has individual sequence.
0	1	1	1	1+1+1	T3(EVT3), T2(EVT2), T1(EVT1)	Three triggers based on event as shown, fourth word is not used.
0	0	1	1	1+1	T2(EVT2), T1(EVT1)	Two triggers based on trigger event as shown.
0	0	0	1	1	T1(EVT1)	One trigger is used based on Event 1.
1	1	1	9	1+1+2	T3(EVT4), T2(EVT3), T1(EVT1, EVT2)	Two of the events are used together and other two triggers are independent triggers. Create up-to 3 possible event triggers.
0	1	1	9	1+2	T2(EVT3), T1(EVT1, EVT2)	Two triggers. One independent trigger and the other on event 1 and event 2 occurring.
0	0	1	9	2	T1(EVT1, EVT2)	Single trigger based on event 1 followed by event 2.
1	9	1	9	2+2	T2(EVT3, EVT4), T1(EVT1, EVT2)	Two pairs of events independently checked as two o/p sequences are correspondingly triggered independently. Event order applicable only between consecutive numbers 1 and 2.
1	1	9	9	1+3	T2(EVT4), T1(EVT1, EVT2, EVT3)	Three trigger events are linked and can be checked in order 1, 2, then 3. The fourth event word is independent. Thus two possible triggers.
0	1	9	9	3	T1(EVT1, EVT2, EVT3)	3 trigger events
1	9	9	9	4	T1 (EVT1, EVT2, EVT3, EVT4)	All 4 events are indicated in consecutive increasing order. Events are checked in order.
Any other combination				Sequence as per configuration	Events linked is 0x9	In case 0x1, the trigger word can cause the trigger. In case 0x9, the trigger word can relay the event to next event in order.

Note

Multiword triggers always look for specific order of events and event words hence strictly the same order needs to be followed. Same sets of events occurring but not in the given order does not cause the trigger.

The SSS does not have to be used, and output can be driven by WADI block instead. However both settings are not to be used simultaneously.

19.5 Lock and Commit Registers

To protect the configuration registers of the WADI and SSS from rouge/run-away code the lock and commit registers are defined. Each of these registers allow range of configuration registers to be protected from writing. Read does not have a restriction. Table 19-6 lists the registers.

Table 19-6. Lock and Commit Registers

Register Name	Protected Address Range	Protected Registers	Comment
CFGREGLOCK	WADI_CONFIG_REGS: 0x000-0xFFF	All registers except Status registers like BLKERRSTS	When set, locks the writes to these registers. Can be unlocked before writes, if commit is not set to 1.
CFGREGCOMMIT	WADI_CONFIG_REGS: 0x000-0xFFF	All registers except Status registers like BLKERRSTS	Once set, does not allow to clear the CONFIG_REG_LOCK set value to 1.
OPERREGLOCK	WADI_OPER_SSS_REGS: 0x100-0x2FF	All registers in this range.	When set, locks the writes to these registers. Can be unlocked before writes, if commit is not set to 1.
OPERREGCOMMIT	WADI_OPER_SSS_REGS: 0x100-0x2FF	All registers in this range.	Once set, does not allow to clear the OPER_REG_LOCK set value to 1.

19.6 Interrupt and Error Handling

Interrupt Generation

Interrupts are generated based in the masked interrupt status of the WADI. Upon the error event, WADI generates an active-low pulse of 3 clock cycles to be triggered to CPU as an interrupt. Any overlapping events during this 3 clock assertion are not separately asserted. If the event gets created at the end of third clock, then next assertion is done from clock 6 covering any other event occurrence until that point

Error Status Handling

Often the signals to be characterized do not have the format or have lag that can affect the measurement counter, for example if measuring pulse width and signal gets stuck at that polarity after first or first few toggles. Or while averaging/accumulating number of counts do not complete and signal gets stuck in active pulse state hence overrunning measurements. Such situations can trigger an event that can assert an interrupt to CPU.BLKERRSTS[SIG_ERR] tracks this and corresponding interrupt registers can be used by CPU to service interrupt.

Once an event is detected, the event is reported to the system either through interrupt event, RTDMA event, and WADI outputs to trigger an event, error status for software debug, or the event gets connected to event trigger for safe state sequencer.

Upon interrupted, the CPU knows which WADI block caused an error and what was the error. The masked interrupt status provides the WADI blocks that triggered error. For each WADI block there is a register that provides the status of which type of event triggered the interrupt. This is stored within BLKERRINFO that can further lead to the exact event type that triggered the error.

The BLKERRINFO register contains information about the first recorded failed count of the signal analysis (BLKERRINFO[ERRNT]) and the type of error (BLKERRINFO[ERRTYPE]).

Note

BLKERRSTS is writable to allow debug and emulation of events during software development. Hence, writes to these registers can trigger events downwards to be an interrupt or RTDMA or SSS.

BLKERRCFG[OVERIDESIGn] are settings used for WADI instances that do not incorporate the SSS. In case the number of event words and sequence words are 0, then OVERIDESIGn give limited flexibility to build the fixed safe states. In cases where SSS components are instantiated, these fields are treated as reserved.

19.7 RTDMA Interfaces

WADI raises RTDMA request based on the events occurring in the WADI block or SSS. WADI has a RTDMA mechanism that has only one RTDMA request and corresponding RTDMA ACK. Hence, it is expected that software configuring RTDMA action knows the range it needs to configure and change WADI settings or operational registers. To make it easier for software to affect only configurations that are affected, the feature for filtered writes is enabled. When writes are filtered, WADI instance tracks the WADI block, events, output events that has an error event and selectively allows change in that WADI block alone. Filtered writes are to the WADI blocks, event words and output events those are enabled for filtered writes.

If the filtered write is not enabled then all writes are allowed to WADI block. When Filtered write is enabled, then any writes within a window of a RTDMA request to RTDMA acknowledgment go through only if there is occurrence of the event within WADI block or event is triggered or output event was active and corresponding event was enabled for RTDMA trigger through DMATRIGEN.

Note

Filtered access rule is applicable only for the window when RTDMA request of WADI is active. At all other times accesses to registered are allowed as long as those are not locked.

For filtered writes within each WADI block there are three types of events and there are dedicated registers enabled for filtered writes. Thus the filtered access are granular based on events internal to the WADI. While the common filtered_en signal and RTDMA trigger enable, RTDMA update status, and RTDMA trigger status is applicable to entire WADI block.

In case the block needs an update regardless of any event, then the block must be left disabled for filtered writes.

If the event occurs after the RTDMA request and RTDMA write is not passed the corresponding address region yet then incoming filtered writes are exercised on the region though originally corresponding RTDMA request did not trigger the writes. Upon assertion of RTDMA acknowledgment the status of the filtered regions which are written is updated to the DMASTSUPDATE. Note DMASTSUPDATE is reserved for filtered writes hence regions unfiltered if written do not reflect here as software has way to deduce the same based on the DMAFILTWRCFG. Upon RTDMA acknowledgment the blocks which were written successfully deassert the RTDMA trigger (DMATRIGSTS).

The following describes the specific WADI registers having DMA filtered writes enabled

- WADI Block registers related DMA filtered writes
  - SIGTOSIGCFG
  - SIGTOSIG_PKCFG
  - SIGTOSIG_AVGCFG
  - SIGTOSIG_DBOLAPA
  - SIGTOSIG_DBOLAPB
  - SIG1/2CMPA
  - SIG1/2CMPB
  - SIG1/2PKCFG
  - SIG1/2AVGCFG
- WADI SSS registers related DMA filtered writes
  - SSS_EVT[1..8]CFG
  - SSS_TRIGEVT1_4CFG
  - SSS_TRIGEVT5_8CFG
  - SSS_BLKSOUTEVT[1..8]CFG
  - SSS_BLK1_2OUTSEL → SSS_BLK7_8OUTSEL
  - SSS_BLKSOUTEVT[1..8]CFG
  - SSS_OUTEVT[1..8]TRIGCFG
  - SSS_OUTEVT[1..8]DUR

19.7.1 RTDMA Trigger

RTDMA Trigger Enable

DMATRIGEN register is used to enable a specific block event or the SSS trigger event or the output events to cause the RTDMA activation. This allows the RTDMA request be selectively raised. DMATRIGEN is the enable for triggering RTDMA, if the event happens hence updates DMATRIGSTS only when corresponding DMATRIGEN.

RTDMA Trigger Status

DMATRIGSTS is a status register that reflects the event that drives the RTDMA request. The status shall remain asserted.

- Either until RTDMA request is serviced and RTDMA ACK for the corresponding event has been updated.
- User clears the status bits with specific 0x0 write to the bit. The hardware update has priority over software writes to avoid loosing the event/ RTDMA action.

RTDMA Update Status

WADI RTDMA can work to update all the configured MMRs without filter or with specific filtered updates to WADI blocks or event words or output events that have triggered the event. Further once the MMR for specific filtered WADI block, event or sequencer is updated the status of the same is recorded in the RTDMA update status register. This status can be cleared by the user writing 0x0 to clear specific bit. Note RTDMA update status is for filtered writes within RTDMA request assertion and RTDMA acknowledgment assertion. DMASTSUPDATE is thus dependent on RTDMA filtered write or occurrence of event.

RTDMA Filtered Write Enable Control

DMAFILTWRCFG register controls settings of filtered writes to be enabled by the RTDMA. When this setting is not enabled the writes to WADI configurations go through without filter. The Read from RTDMA access have no restrictions.

WADI has no means to track whether access is from RTDMA or CPU; hence, any access after RTDMA request assertion and before RTDMA acknowledgment is filtered to WADI block for which it is enabled.

19.8 Software

19.8.1 WADI Registers to Driverlib Functions

Table 19-7. WADI Registers to Driverlib Functions

File	Driverlib Function
BLKCFG	
-	
SIGTOSIGCFG	
-	
SIGTOSIG_PKCFG	
-	
SIGTOSIG_AVGCFG	
-	
SIGTOSIG_DBOLAPA	
-	
SIGTOSIG_DBOLAPB	
-	
BLKTRIGCFG	
-	

Table 19-7. WADI Registers to Driverlib Functions (continued)

File	Driverlib Function
SIG1CFG	
-	
SIG1CMPA	
-	
SIG1CMPB	
-	
SIG1PKCFG	
-	
SIG1AVGCFG	
-	
SIG1EDGECFG	
-	
SIG1EDGEMVWCFG	
-	
SIG2CFG	
-	
SIG2CMPA	
-	
SIG2CMPB	
-	
SIG2PKCFG	
-	
SIG2AVGCFG	
-	
SIG2EDGECFG	
-	
SIG2EDGEMVWCFG	
-	
BLKERRSTS	
-	
BLKERRINFO	
-	
BLKERRCFG	
-	
SSS_EVTMASK	
-	
PARTEST	
-	
BASETIMERLOW	
-	
BASETIMERHIGH	
-	
INTSTS	
-	
INTSTSMASK	

Table 19-7. WADI Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
BLKSMASKSTS	
-	
INTSTSCLR	
-	
INTSTSFRC	
-	
SIGSYNCFILTCFG	
-	
TRIGSYNCFILTCFG	
-	
REVISION	
-	
DMATRIGSTS	
-	
DMATRIGEN	
-	
DMASTSUPDATE	
-	
DMAFILTWRCFG	
-	
CFGREGLOCK	
-	
CFGREGCOMMIT	
-	
OPERREGLOCK	
-	
OPERREGCOMMIT	
-	
SSS_EVTTRIG	
-	
SSS_OUTEVTSTS	
-	
SSS_BLK1_2OUTSEL	
-	
SSS_BLK3_4OUTSEL	
-	
SSS_BLK5_6OUTSEL	
-	
SSS_BLK7_8OUTSEL	
-	
SSS_OUTEVT1LINKCFG	
-	
SSS_OUTEVT2LINKCFG	
-	

Table 19-7. WADI Registers to Driverlib Functions (continued)

File	Driverlib Function
SSS_OUTEVT3LINKCFG	
-	
SSS_OUTEVT4LINKCFG	
-	
SSS_OUTEVT5LINKCFG	
-	
SSS_OUTEVT6LINKCFG	
-	
SSS_OUTEVT7LINKCFG	
-	
SSS_OUTEVT8LINKCFG	
-	
SSS_EVT1CFG	
-	
SSS_EVT2CFG	
-	
SSS_EVT3CFG	
-	
SSS_EVT4CFG	
-	
SSS_TRIGEVT1_4CFG	
-	
SSS_BLKSOUTEVT1CFG	
-	
SSS_BLKSOUTEVT2CFG	
-	
SSS_BLKSOUTEVT3CFG	
-	
SSS_BLKSOUTEVT4CFG	
-	
SSS_OUTEVT1TRIGCFG	
-	
SSS_OUTEVT2TRIGCFG	
-	
SSS_OUTEVT3TRIGCFG	
-	
SSS_OUTEVT4TRIGCFG	
-	
SSS_OUTEVT1DUR	
-	
SSS_OUTEVT2DUR	
-	
SSS_OUTEVT3DUR	
-	
SSS_OUTEVT4DUR	

Table 19-7. WADI Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
SSS_EVT5CFG	
-	
SSS_EVT6CFG	
-	
SSS_EVT7CFG	
-	
SSS_EVT8CFG	
-	
SSS_TRIGEVT5_8CFG	
-	
SSS_BLKSOUTEVT5CFG	
-	
SSS_BLKSOUTEVT6CFG	
-	
SSS_BLKSOUTEVT7CFG	
-	
SSS_BLKSOUTEVT8CFG	
-	
SSS_OUTEVT5TRIGCFG	
-	
SSS_OUTEVT6TRIGCFG	
-	
SSS_OUTEVT7TRIGCFG	
-	
SSS_OUTEVT8TRIGCFG	
-	
SSS_OUTEVT5DUR	
-	
SSS_OUTEVT6DUR	
-	
SSS_OUTEVT7DUR	
-	
SSS_OUTEVT8DUR	
-	
PARTEST	
-	

19.8.2 WADI Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/wadi

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

19.8.2.1 WADI Duty and Frequency check - SINGLE_CORE

FILE: wadi_ex1_duty_freq_capture.c

This example configures WADI1 Block 1 and Block 2 as follows

- WADI1 Block 1 has an input from the PWM using INPUTXBAR
- Pulse width checking is enabled with defined margin of error
- WADI1 Block 2 has an input from the PWM using INPUTXBAR
- Frequency checking is enabled with defined margin of error

The WADI block has error status register that can be used to monitor if any erroneous signals occur. (BLKERRSTS)

This example also makes use of the Input X-BAR. GPIO0 (ePWM1A output) is routed to the input X-BAR, from which it is routed to WADI SIG1 input.

19.9 WADI Registers

This Section describes the WADI Registers.

19.9.1 WADI Base Address Table

Table 19-8. WADI Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
WADI_CONFIG_REGS	WADI1BLK1CONFIG_BASE	0x600B_0000	YES	YES	YES	YES	YES	YES	-	YES
WADI_CONFIG_REGS	WADI1BLK2CONFIG_BASE	0x600B_0100	YES	YES	YES	YES	YES	YES	-	YES
WADI_CONFIG_REGS	WADI1BLK3CONFIG_BASE	0x600B_0200	YES	YES	YES	YES	YES	YES	-	YES
WADI_CONFIG_REGS	WADI1BLK4CONFIG_BASE	0x600B_0300	YES	YES	YES	YES	YES	YES	-	YES
WADI_OPER_SSS_REGS	WADI1OPERSSS_BASE	0x600B_1000	YES	YES	YES	YES	YES	YES	-	YES
WADI_CONFIG_REGS	WADI2BLK1CONFIG_BASE	0x600B_2000	YES	YES	YES	YES	YES	YES	-	YES
WADI_CONFIG_REGS	WADI2BLK2CONFIG_BASE	0x600B_2100	YES	YES	YES	YES	YES	YES	-	YES
WADI_CONFIG_REGS	WADI2BLK3CONFIG_BASE	0x600B_2200	YES	YES	YES	YES	YES	YES	-	YES
WADI_CONFIG_REGS	WADI2BLK4CONFIG_BASE	0x600B_2300	YES	YES	YES	YES	YES	YES	-	YES
WADI_OPER_SSS_REGS	WADI2OPERSSS_BASE	0x600B_3000	YES	YES	YES	YES	YES	YES	-	YES

19.9.2 WADI_CONFIG_REGS Registers

Table 19-9 lists the memory-mapped registers for the WADI_CONFIG_REGS registers. All register offset addresses not listed in Table 19-9 should be considered as reserved locations and the register contents should not be modified.

Table 19-9. WADI_CONFIG_REGS Registers

Offset	Acronym	Register Name	Protection
0h	BLKCFG	WADI Block Clock Divider	PARITY_PROTECTED
4h	SIGTOSIGCFG	Configuration for SIG1 to SIG2 comparison for WADI Block	PARITY_PROTECTED
8h	SIGTOSIG_PKCFG	Signal1 to Signal2 peak comparison value and margin for WADI Block	PARITY_PROTECTED
Ch	SIGTOSIG_AVGCFG	Signal1 to Signal2 average comparison value and margin for WADI Block	PARITY_PROTECTED
10h	SIGTOSIG_DBOLAPA	Signal1 to Signal2 dead-band comparison difference and margin for WADI Block	PARITY_PROTECTED
14h	SIGTOSIG_DBOLAPB	Signal1 to Signal2 dead-band and Overlap compare values for WADI Block	PARITY_PROTECTED
18h	BLKTRIGCFG	The trigger configuration for SIG1 and SIG2 of the WADI Block	PARITY_PROTECTED
40h	SIG1CFG	Measurement configuration for SIG1 of WADI Block	PARITY_PROTECTED
44h	SIG1CMPA	Envelope-1 compare value and margin for SIG1 for WADI Block	PARITY_PROTECTED
48h	SIG1CMPB	Envelope-2 compare value and margin for SIG1 for WADI Block	PARITY_PROTECTED
4Ch	SIG1PKCFG	Peak of the aggregation compare configuration for SIG1 of WADI Block	PARITY_PROTECTED
50h	SIG1AVGCFG	Average of the aggregation comparison configuration for SIG1 of WADI Block	PARITY_PROTECTED
54h	SIG1EDGECFG	Frequency measurement configuration for SIG1 of WADI Block	PARITY_PROTECTED
58h	SIG1EDGEMVWCFG	Moving Window Configuration for SIG1 of WADI Block	PARITY_PROTECTED
80h	SIG2CFG	Measurement configuration for SIG2 of WADI Block	PARITY_PROTECTED
84h	SIG2CMPA	Compare value A and +/- margin for SIG2 for WADI Block	PARITY_PROTECTED
88h	SIG2CMPB	Compare value B and +/- margin for SIG2 for WADI Block	PARITY_PROTECTED
8Ch	SIG2PKCFG	Peak of the aggregation compare configuration for SIG2 of WADI Block	PARITY_PROTECTED
90h	SIG2AVGCFG	Average of the aggregation comparison configuration for SIG2 of WADI Block	PARITY_PROTECTED
94h	SIG2EDGECFG	Frequency measurement configuration for SIG2 of WADI Block	PARITY_PROTECTED
98h	SIG2EDGEMVWCFG	Moving Window Configuration for SIG2 of WADI Block	PARITY_PROTECTED
C0h	BLKERRSTS	Block status register	
C4h	BLKERRINFO	Block count value of failed register	
C8h	BLKERRCFG	Block Debug additional info.	PARITY_PROTECTED
CCh	SSS_EVTMASK	Mask to combine Block errors for trigger of SSS in event word Output by WADI block	PARITY_PROTECTED

Table 19-9. WADI_CONFIG_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
E8h	PARTEST	Enables parity test	

Complex bit access types are encoded to fit into small table cells. Table 19-10 shows the codes that are used for access types in this section.

Table 19-10. WADI_CONFIG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

19.9.2.1 BLKCFG Register (Offset = 0h) [Reset = 00000000h]

BLKCFG is shown in Figure 19-13 and described in Table 19-11.

Return to the Summary Table.

Input Clock Divider control for the block

Figure 19-13. BLKCFG Register

31	30	29	28	27	26	25	24
TRIG2IN				TRIG1IN			
R/W-0h				R/W-0h			
23	22	21	20	19	18	17	16
RESERVED			SIG2IN				
R-0h			R/W-0h				
15	14	13	12	11	10	9	8
RESERVED			SIG1IN				
R-0h			R/W-0h				
7	6	5	4	3	2	1	0
RESERVED			CLKEN	RESERVED	RESERVED		
R-0h			R/W-0h	R-0h	R/W-0h		

Table 19-11. BLKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	TRIG2IN	R/W	0h	Mux select value for cross-bar to select one of the input of TRIG2 Refer to configuration input table Reset type: SYSRSn
27-24	TRIG1IN	R/W	0h	Mux select value for cross-bar to select one of the input of TRIG1 Refer to configuration input table Reset type: SYSRSn
23-21	RESERVED	R	0h	Reserved
20-16	SIG2IN	R/W	0h	Mux select value for cross-bar to select one of the input of SIG2 Refer to configuration input table Reset type: SYSRSn
15-13	RESERVED	R	0h	Reserved
12-8	SIG1IN	R/W	0h	Mux select value for cross-bar to select one of the input of SIG1 Refer to configuration input table Reset type: SYSRSn
7-5	RESERVED	R	0h	Reserved
4	CLKEN	R/W	0h	0: Disable clock 1: Enable clock Reset type: SYSRSn
3	RESERVED	R	0h	Reserved
2-0	RESERVED	R/W	0h	Reserved

19.9.2.2 SIGTOSIGCFG Register (Offset = 4h) [Reset = 00000000h]

SIGTOSIGCFG is shown in Figure 19-14 and described in Table 19-12.

Return to the Summary Table.

Configuration for SIG1 to SIG2 comparison for WADI Block

Figure 19-14. SIGTOSIGCFG Register

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

Table 19-12. SIGTOSIGCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Reserved
29	OLAPCMPTYPE	R/W	0h	When configured selects type of OLAP check as below. Default is configured edge 0: Overlap of levels per configuration 1: Overlap of levels per configuration and complementary levels as well Ex: 0: Overalap of SIG1-level-High and SIG2-level-Low 1: Overalap of SIG1-level-High and SIG2-level-Low and Overalap of SIG1-level-Low and SIG2-level-High Reset type: SYSRSn
28	DBCMPTYPE	R/W	0h	When configured selects type of comparison for deadband as below. Default is configured edge 0: Deadband of configured edge comparison 1: Deadband of configured and opposite edge pairs comparison Ex: SIG1CFG[SIGPOL] for rise to SIG2CFG[SIGPOL] for fall 0: Deadband between asswertion of SIG1 rise and SIG2 fall 1: Deadband between assertion of SIG1 rise and SIG2 fall, and assertion of SIG1 fall and SIG2 rise Reset type: SYSRSn
27	SIGTOSIG_OLAPCMPEN	R/W	0h	When set enables the overlap comparison For overlap signal edge polarity is considered for measuring level post edge assertion. e.g. rise edge-high level Reset type: SYSRSn
26	SIGTOSIG_DBCMPEN	R/W	0h	When set enables the deadband comparison The distance between configured edges of both signals is compared. Reset type: SYSRSn
25	RESERVED	R	0h	Reserved

Table 19-12. SIGTOSIGCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	SIGTOSIG_CMPEN	R/W	0h	This bit when set enables the SIG1 to SIG2 comparison of measurements either widths or edge counts based on block configuration Default '0' , hence no SIGTOSIG compare Reset type: SYSRSn
23-20	RESERVED	R	0h	Reserved
19-16	CMPMARGIN	R/W	0h	This margin (+/-) against SIGTOSIGCFG[CMP] allows the variation of comparison to be in range for difference of measurements of two signals Reset type: SYSRSn
15-0	CMP	R/W	0h	This is the compare value for difference of SIG1 and SIG2 measurement either width or edge count depending upon the mode set for SIG1 The difference is compared against the base of this value with +/- margin as in CMPMARGIN Reset type: SYSRSn

19.9.2.3 SIGTOSIG_PKCFG Register (Offset = 8h) [Reset = 00000000h]

SIGTOSIG_PKCFG is shown in Figure 19-15 and described in Table 19-13.

Return to the Summary Table.

Signal to Signal peak compare difference and threshold

Figure 19-15. SIGTOSIG_PKCFG Register

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

Table 19-13. SIGTOSIG_PKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	PKCMPMARGIN	R/W	0h	Margin of variation for difference of peak within aggregated widths or edge count of both signals Reset type: SYSRSn
23-0	PKCMP	R/W	0h	Compare difference for peak within aggregated measurements of either widths or edge counts of both signals Reset type: SYSRSn

19.9.2.4 SIGTOSIG_AVGCFG Register (Offset = Ch) [Reset = 00000000h]

SIGTOSIG_AVGCFG is shown in Figure 19-16 and described in Table 19-14.

Return to the Summary Table.

Signal to Signal average compare difference and threshold

Figure 19-16. SIGTOSIG_AVGCFG Register

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

Table 19-14. SIGTOSIG_AVGCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	AVGCMPMARGIN	R/W	0h	Margin of variation of average of aggregated widths or edge counts of both signals Reset type: SYSRSn
23-0	AVGCMP	R/W	0h	Compare value for average of aggregated widths or edge counts of both signals Reset type: SYSRSn

19.9.2.5 SIGTOSIG_DBOLAPA Register (Offset = 10h) [Reset = 00000000h]

SIGTOSIG_DBOLAPA is shown in Figure 19-17 and described in Table 19-15.

Return to the Summary Table.

Signal to Signal dead-band and overlap compare difference and threshold pair-1

Figure 19-17. SIGTOSIG_DBOLAPA Register

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

Table 19-15. SIGTOSIG_DBOLAPA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Reserved
29-24	OLAPCMPAMARGIN	R/W	0h	Margin of variation for overlap between both signals, thus overlap count is compared to be between Compare value +/- Compare margin. Reset type: SYSRSn
23-8	OLAPCMPA	R/W	0h	Compare value A for overlap between both signal levels Reset type: SYSRSn
7-0	DBCMPA	R/W	0h	Compare value A for deadband between both signals Reset type: SYSRSn

19.9.2.6 SIGTOSIG_DBOLAPB Register (Offset = 14h) [Reset = 00000000h]

SIGTOSIG_DBOLAPB is shown in Figure 19-18 and described in Table 19-16.

Return to the Summary Table.

Signal to Signal dead-band and overlap compare difference and threshold pair-2

Figure 19-18. SIGTOSIG_DBOLAPB Register

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

Table 19-16. SIGTOSIG_DBOLAPB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Reserved
29-24	OLAPCMPBMARGIN	R/W	0h	Margin of variation for overlap between both signals, thus overlap count is compared to be between Compare value +/- Compare margin. Reset type: SYSRSn
23-8	OLAPCMPB	R/W	0h	Compare value B for overlap between both signal levels Reset type: SYSRSn
7-0	DBCMPB	R/W	0h	Compare value B for deadband between both signals Reset type: SYSRSn

19.9.2.7 BLKTRIGCFG Register (Offset = 18h) [Reset = 00000000h]

BLKTRIGCFG is shown in Figure 19-19 and described in Table 19-17.

Return to the Summary Table.

The trigger configuration for SIG1 and SIG2 of the WADI Block

Figure 19-19. BLKTRIGCFG Register

31	30	29	28	27	26	25	24
RESERVED							TRIG2SWEN
R-0h							R-0/W1S-0h
23	22	21	20	19	18	17	16
RESERVED					TRIG2TYPE		
R-0h					R/W-0h		
15	14	13	12	11	10	9	8
RESERVED							TRIG1SWEN
R-0h							R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED					TRIG1TYPE		
R-0h					R/W-0h		

Table 19-17. BLKTRIGCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	TRIG2SWEN	R-0/W1S	0h	When this signal is set the measurement for SIG2 is started. It is also used in 'Either-0x3' trigger condition 'TRIG2TYPE' for SIG2. Reset type: SYSRSn
23-19	RESERVED	R	0h	Reserved
18-16	TRIG2TYPE	R/W	0h	Selects the type of trigger that shall be used for the SIG2 signal. 0x0: No Trigger - In this mode the signal will be measured right after configuration is done 0x1: Hardware Trigger , based on the assertion of the trigger the measurement begins 0x2: Software Trigger, after software trigger assertion measurement begins 0x3: Either trigger, the measurement begins when either of the software or hardware trigger is asserted. 0x4: Synchronized Trigger, When Set no trigger setting from SIG2 is used but SIG1 setting is used. If both are set to Synchronized then either SIG1 or SIG2 hardware triggers will start measurement 0x5-0x7: Reserved and defaults to NO trigger condition Reset type: SYSRSn
15-9	RESERVED	R	0h	Reserved
8	TRIG1SWEN	R-0/W1S	0h	When this signal is set the measurement for SIG1 is started. It is also used in 'Either-0x3' trigger condition 'TRIG1TYPE' for SIG1. Reset type: SYSRSn
7-3	RESERVED	R	0h	Reserved

Table 19-17. BLKTRIGCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	TRIG1TYPE	R/W	0h	Selects the type of trigger that shall be used for the SIG1 signal. 0x0: No Trigger - In this mode the signal will be measured right after configuration is done 0x1: Hardware Trigger , based on the assertion of the trigger the measurement begins 0x2: Software Trigger, after software trigger assertion measurement begins 0x3: Either trigger, the measurement begins when either of the software or hardware trigger is asserted. 0x4: Synchronized Trigger, When Set no trigger setting from SIG1 is used but SIG2 setting is used. If both are set to Synchronized then either SIG1 or SIG2 hardware triggers will start measurement 0x5-0x7: Reserved and defaults to NO trigger condition Reset type: SYSRSn

19.9.2.8 SIG1CFG Register (Offset = 40h) [Reset = 00000000h]

SIG1CFG is shown in Figure 19-20 and described in Table 19-18.

Return to the Summary Table.

Edge, width measurement configuration to characterize the SIG1 of WADI block

Figure 19-20. SIG1CFG Register

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

Table 19-18. SIG1CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25-24	AGGRMODE	R/W	0h	Defines if peak or average value of the aggregated set of values if aggregation is enabled. 0x0: Aggregation 0x1: Aggregation & peak 0x2: Aggregation & Avg 0x3: Aggregation & peak & Avg Reset type: SYSRSn
23-19	RESERVED	R	0h	Reserved
18-16	NUMAGGR	R/W	0h	Number of measurements to be accumulated together to compare against aggregate energy. Once the first measurement is over next measurement starts with configured EDGE_TYPE. 0x0: Aggregation is disabled, each measurements are independently compared. 0x1: Aggregation of 2 measurements 0x2: Aggregation of 4 measurements 0x3: Aggregation of 8 measurements 0x4: Aggregation of 16 measurements 0x5: Aggregation of 32 measurements 0x6: Aggregation of 64 measurements 0x7: Reserved (Defaults to Disabled) Reset type: SYSRSn
15-12	RESERVED	R	0h	Reserved

Table 19-18. SIG1CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-8	EDGESPAN	R/W	0h	Defines span from starting edge in terms of number of edges to measure the width in terms of WADI block clock count. 0x0: No measurement 0x1: Smallest pulse on signal 0x2: Smallest period on signal 0x3: Period to third edge from start edge after trigger condition .... 0xF: Period of 15th edge from start edge after trigger condition. Once this nth edge is over one measurement is complete and is used for comparison. Reset type: SYSRSn
7-2	RESERVED	R	0h	Reserved
1-0	SIGPOL	R/W	0h	Selects if rise, fall or either edge should be considered. 0x0 : Reserved (Default Rise) 0x1 : Rise edge 0x2 : Fall edge 0x3 : Either edge Reset type: SYSRSn

19.9.2.9 SIG1CMPA Register (Offset = 44h) [Reset = 00000000h]

SIG1CMPA is shown in Figure 19-21 and described in Table 19-19.

Return to the Summary Table.

Configuration of Envelope-1 compare of the measurement count of width or edges as per configuration for SIG1 of WADI block

Figure 19-21. SIG1CMPA Register

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

Table 19-19. SIG1CMPA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	CMPAMARGIN	R/W	0h	Margin to compare in an envelop of +/- range around SIG1CMPA[CMPA] against the measured count of the SIG1 Reset type: SYSRSn
23-0	CMPA	R/W	0h	Compare A value against the measured count for width or frequency of the SIG1. The measured count accumulation is based upon the SIG1CFG Reset type: SYSRSn

19.9.2.10 SIG1CMPB Register (Offset = 48h) [Reset = 00000000h]

SIG1CMPB is shown in Figure 19-22 and described in Table 19-20.

Return to the Summary Table.

Configuration of Envelope-2 compare of the measurement count of width or edges as per configuration for SIG1 of WADI block

Figure 19-22. SIG1CMPB Register

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

Table 19-20. SIG1CMPB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	CMPBMARGIN	R/W	0h	Margin to compare in an envelop of +/- range around SIG1CMPB[CMPB] against the measured count of the SIG1 Reset type: SYSRSn
23-0	CMPB	R/W	0h	Compare B value against the measured count for width or frequency of the SIG1. The measured count accumulation is based upon the SIG1CFG Reset type: SYSRSn

19.9.2.11 SIG1PKCFG Register (Offset = 4Ch) [Reset = 00000000h]

SIG1PKCFG is shown in Figure 19-23 and described in Table 19-21.

Return to the Summary Table.

Compare value and margin for the peak of the aggregation for SIG1 of WADI Block

Figure 19-23. SIG1PKCFG Register

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

Table 19-21. SIG1PKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	CMPMARGIN	R/W	0h	Margin to compare +/- around SIG1CMPB[CMP] and compared against the detected peak out of aggregated measurements- of the SIG1 Reset type: SYSRSn
23-0	CMP	R/W	0h	Compare value against detected peak out of aggregated measurements for SIG1 Reset type: SYSRSn

19.9.2.12 SIG1AVGCFG Register (Offset = 50h) [Reset = 00000000h]

SIG1AVGCFG is shown in Figure 19-24 and described in Table 19-22.

Return to the Summary Table.

Compare value and margin for the average of the aggregation for SIG1 of WADI Block

Figure 19-24. SIG1AVGCFG Register

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

Table 19-22. SIG1AVGCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	CMPMARGIN	R/W	0h	Margin to compare +/- around SIG1AVGCMP[CMP] and compared against the average of aggregated measurements- of the SIG1 Reset type: SYSRSn
23-0	CMP	R/W	0h	Compare value against average of aggregated measurements for SIG1 Reset type: SYSRSn

19.9.2.13 SIG1EDGECFG Register (Offset = 54h) [Reset = 00000000h]

SIG1EDGECFG is shown in Figure 19-25 and described in Table 19-23.

Return to the Summary Table.

Enable, configurations, window size settings for SIG1 of WADI block

Figure 19-25. SIG1EDGECFG Register

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

Table 19-23. SIG1EDGECFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CNTEN	R/W	0h	When set high the edge count measurement is enabled Measurements accumulated are of the type edge count SIG1CFG[SIGPOL] is used for measuring type of edges Reset type: SYSRSn
30-28	RESERVED	R	0h	Reserved
27-0	TIMEWNDOW	R/W	0h	This is time window count used for tracking global time count (in terms of system clock) within which the edges of input waveform of the configured EDGE_TYPE are counted Reset type: SYSRSn

19.9.2.14 SIG1EDGEMVWCFG Register (Offset = 58h) [Reset = 00000000h]

SIG1EDGEMVWCFG is shown in Figure 19-26 and described in Table 19-24.

Return to the Summary Table.

Configuration of moving window for SIG1 frequency measurement of WADI Block

Figure 19-26. SIG1EDGEMVWCFG Register

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

Table 19-24. SIG1EDGEMVWCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MVWEN	R/W	0h	This bit when set enables the moving time window feature. Reset type: SYSRSn
30	RESERVED	R	0h	Reserved
29-28	MVWCNT	R/W	0h	Number of moving time window Snapshots taken. Multiple snapshots complete one Window of frequency measurement. 0x0 : Only one snapshot for window count. No moving window measurements 0x1 : Two snapshots in one window. The reading is for fixed window, despite size difference there is no moving operation. 0x2 : Three snapshots in one window it means Window is bisected. 0x3 : Four snapshots in one window it means Fixed window has three moving window portions. Reset type: SYSRSn
27-26	RESERVED	R	0h	Reserved
25-0	MVWTIME	R/W	0h	Moving window time provides the smaller unit such that multiple MVW can be counted for completing one reading. And advancement of each such count gives full window reading at smaller than a window increments. This time is directly used to count the portion of window. With MVWCNT iterations of MVWTIME measurements one can get the full window reading. Every MVWTIME measurement is recorded and readings of multiple part used to get reading of full window, hence at every MVWTIME interval, the earliest reading of equivalent period is retired and recent reading for equivalent period is added. Thus giving moving window measurement check. Reset type: SYSRSn

19.9.2.15 SIG2CFG Register (Offset = 80h) [Reset = 00000000h]

SIG2CFG is shown in Figure 19-27 and described in Table 19-25.

Return to the Summary Table.

Edge, width measurement configuration to characterize the SIG2 of WADI block

Figure 19-27. SIG2CFG Register

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

Table 19-25. SIG2CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25-24	AGGRMODE	R/W	0h	Defines if peak or average value of the aggregated set of values if aggregation is enabled. 0x0: Aggregation 0x1: Aggregation & peak 0x2: Aggregation & Avg 0x3: Aggregation & peak & Avg Reset type: SYSRSn
23-19	RESERVED	R	0h	Reserved
18-16	NUMAGGR	R/W	0h	Number of measurements to be accumulated together to compare against aggregate energy. Once the first measurement is over next measurement starts with configured EDGE_TYPE. 0x0: Aggregation is disabled, each measurements are independently compared. 0x1: Aggregation of 2 measurements 0x2: Aggregation of 4 measurements 0x3: Aggregation of 8 measurements 0x4: Aggregation of 16 measurements 0x5: Aggregation of 32 measurements 0x6: Aggregation of 64 measurements 0x7: Reserved (Defaults to Disabled) Reset type: SYSRSn
15-12	RESERVED	R	0h	Reserved

Table 19-25. SIG2CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-8	EDGESPAN	R/W	0h	Defines span from starting edge in terms of number of edges to measure the width in terms of WADI block clock count. 0x0: No measurement 0x1: Smallest pulse on signal 0x2: Smallest period on signal 0x3: Period to third edge from start edge after trigger condition .... 0xF: Period of 15th edge from start edge after trigger condition. Once this nth edge is over one measurement is complete and is used for comparison. Reset type: SYSRSn
7-2	RESERVED	R	0h	Reserved
1-0	SIGPOL	R/W	0h	Selects if rise, fall or either edge should be considered. 0x0 : Reserved (Default Rise) 0x1 : Rise edge 0x2 : Fall edge 0x3 : Either edge Reset type: SYSRSn

19.9.2.16 SIG2CMPA Register (Offset = 84h) [Reset = 00000000h]

SIG2CMPA is shown in Figure 19-28 and described in Table 19-26.

Return to the Summary Table.

Configuration of Envelope-1 compare of the measurement count of width or edges as per configuration for SIG2 of WADI block

Figure 19-28. SIG2CMPA Register

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

Table 19-26. SIG2CMPA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	CMPAMARGIN	R/W	0h	Margin to compare in an envelop of +/- range around SIG2CMPA[CMPA] against the measured count of the SIG2 Reset type: SYSRSn
23-0	CMPA	R/W	0h	Compare A value against the measured count for width or frequency of the SIG2. The measured count accumulation is based upon the SIG2CFG Reset type: SYSRSn

19.9.2.17 SIG2CMPB Register (Offset = 88h) [Reset = 00000000h]

SIG2CMPB is shown in Figure 19-29 and described in Table 19-27.

Return to the Summary Table.

Configuration of Envelope-2 compare of the measurement count of width or edges as per configuration for SIG2 of WADI block

Figure 19-29. SIG2CMPB Register

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

Table 19-27. SIG2CMPB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	CMPBMARGIN	R/W	0h	Margin to compare in an envelop of +/- range around SIG2CMPB[CMPB] against the measured count of the SIG2 Reset type: SYSRSn
23-0	CMPB	R/W	0h	Compare B value against the measured count for width or frequency of the SIG2. The measured count accumulation is based upon the SIG2CFG Reset type: SYSRSn

19.9.2.18 SIG2PKCFG Register (Offset = 8Ch) [Reset = 00000000h]

SIG2PKCFG is shown in Figure 19-30 and described in Table 19-28.

Return to the Summary Table.

Compare value and margin for the peak of the aggregation for SIG2 of WADI Block

Figure 19-30. SIG2PKCFG Register

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

Table 19-28. SIG2PKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	CMPMARGIN	R/W	0h	Margin to compare +/- around SIG2CMPB[CMP] and compared against the detected peak out of aggregated measurements- of the SIG2 Reset type: SYSRSn
23-0	CMP	R/W	0h	Compare value against detected peak out of aggregated measurements for SIG2 Reset type: SYSRSn

19.9.2.19 SIG2AVGCFG Register (Offset = 90h) [Reset = 00000000h]

SIG2AVGCFG is shown in Figure 19-31 and described in Table 19-29.

Return to the Summary Table.

Compare value and margin for the average of the aggregation for SIG2 of WADI Block

Figure 19-31. SIG2AVGCFG Register

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

Table 19-29. SIG2AVGCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	CMPMARGIN	R/W	0h	Margin to compare +/- around SIG2AVGCMP[CMP] and compared against the average of aggregated measurements- of the SIG2 Reset type: SYSRSn
23-0	CMP	R/W	0h	Compare value against average of aggregated measurements for SIG2 Reset type: SYSRSn

19.9.2.20 SIG2EDGECFG Register (Offset = 94h) [Reset = 00000000h]

SIG2EDGECFG is shown in Figure 19-32 and described in Table 19-30.

Return to the Summary Table.

Enable, configurations, window size settings for SIG2 of WADI block

Figure 19-32. SIG2EDGECFG Register

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

Table 19-30. SIG2EDGECFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CNTEN	R/W	0h	When set high the edge count measurement is enabled Measurements accumulated are of the type edge count SIG2CFG[SIGPOL] is used for measuring type of edges Reset type: SYSRSn
30-28	RESERVED	R	0h	Reserved
27-0	TIMEWNDOW	R/W	0h	This is time window count used for tracking global time count (in terms of system clock) within which the edges of input waveform of the configured EDGE_TYPE are counted Reset type: SYSRSn

19.9.2.21 SIG2EDGEMVWCFG Register (Offset = 98h) [Reset = 00000000h]

SIG2EDGEMVWCFG is shown in Figure 19-33 and described in Table 19-31.

Return to the Summary Table.

Configuration of moving window for SIG2 frequency measurement of WADI Block

Figure 19-33. SIG2EDGEMVWCFG Register

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

Table 19-31. SIG2EDGEMVWCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MVWEN	R/W	0h	This bit when set enables the moving time window feature. Reset type: SYSRSn
30	RESERVED	R	0h	Reserved
29-28	MVWCNT	R/W	0h	Number of moving time window Snapshots taken. Multiple snapshots complete one Window of frequency measurement. 0x0 : Only one snapshot for window count. No moving window measurements 0x1 : Two snapshots in one window. The reading is for fixed window, despite size difference there is no moving operation. 0x2 : Three snapshots in one window it means Window is bisected. 0x3 : Four snapshots in one window it means Fixed window has three moving window portions. Reset type: SYSRSn
27-26	RESERVED	R	0h	Reserved
25-0	MVWTIME	R/W	0h	Moving window time provides the smaller unit such that multiple MVW can be counted for completing one reading. And advancement of each such count gives full window reading at smaller than a window increments. This time is directly used to count the portion of window. With MVWCNT iterations of MVWTIME measurements one can get the full window reading. Every MVWTIME measurement is recorded and readings of multiple part used to get reading of full window, hence at every MVWTIME interval, the earliest reading of equivalent period is retired and recent reading for equivalent period is added. Thus giving moving window measurement check. Reset type: SYSRSn

19.9.2.22 BLKERRSTS Register (Offset = C0h) [Reset = 00000000h]

BLKERRSTS is shown in Figure 19-34 and described in Table 19-32.

Return to the Summary Table.

Block status register.

Figure 19-34. BLKERRSTS Register

31	30	29	28	27	26	25	24
SIGTOSIG_ERR	SIG_ERR	RESERVED		SIGTOSIG_OLAPCMPBERR		SIGTOSIG_OLAPCMPAERR	
R/W-0h	R/W-0h	R-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
SIGTOSIG_DBCMPBERR	SIGTOSIG_DBCMPAERR	SIGTOSIG_AVGERR		SIGTOSIG_PKERR		SIGTOSIG_CMPERR	
R/W-0h	R/W-0h	R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
SIG2CMPBERR		SIG2AVGERR		SIG2PKERR		SIG2CMPAERR	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
SIG1CMPBERR		SIG1AVGERR		SIG1PKERR		SIG1CMPAERR	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 19-32. BLKERRSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SIGTOSIG_ERR	R/W	0h	When SIGTOSIG comparisons is enabled and one of the signal has accumulated second reading while other signal has not finished even the first reading then this error is set. This could also occur at the start of the sequence hence application software needs to evaluate the condition before action. 0x0 : No Error 0x1 : Error with one of the signal lagging other. Reset type: SYSRSn
30	SIG_ERR	R/W	0h	This bit aggregates error from SIG1 & SIG2 when measurements cannot be completed due to Anomalies such as signal being stuck to value , no toggles etc. 0x0 : No Error 0x1 : Error in measurement. Reset type: SYSRSn
29-28	RESERVED	R	0h	Reserved
27-26	SIGTOSIG_OLAPCMPBERR	R/W	0h	This status indicates the error related to the overlap Compare B check between two signals 0x0 : No Error 0x1 : Reserved (No Error Default) 0x2 : Error violating Max limit 0x3 : Error violating Min Limit Reset type: SYSRSn
25-24	SIGTOSIG_OLAPCMPAERR	R/W	0h	This status indicates the error related to the overlap Compare A check between two signals 0x0 : No Error 0x1 : Reserved (No Error Default) 0x2 : Error violating Max limit 0x3 : Error violating Min Limit Reset type: SYSRSn

Table 19-32. BLKERRSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	SIGTOSIG_DBCMPBERR	R/W	0h	This status indicates the error related to the dead-band between two signals 0x0 : No Error 0x1 : Dead-band Compare B value is violated. Reset type: SYSRSn
22	SIGTOSIG_DBCMPAERR	R/W	0h	This status indicates the error related to the dead-band between two signals 0x0 : No Error 0x1 : Dead-band Compare A value is violated. Reset type: SYSRSn
21-20	SIGTOSIG_AVGERR	R/W	0h	This status indicates the error related to Average of aggregated count is out of range for SIGTOSIG difference 0x0 : No Error 0x1 : Reserved (No Error Default) 0x2 : Error violating Max limit 0x3 : Error violating Min Limit Reset type: SYSRSn
19-18	SIGTOSIG_PKERR	R/W	0h	This status indicates the error related to the difference of peak within aggregated values of SIG1 and SIG2 measurements 0x0 : No Error 0x1 : Reserved (No Error Default) 0x2 : Error violating Max limit 0x3 : Error violating Min Limit Reset type: SYSRSn
17-16	SIGTOSIG_CMPERR	R/W	0h	This status indicates the error related to the difference count is out of range of SIG1 and SIG2 measurements 0x0 : No Error 0x1 : Reserved (No Error Default) 0x2 : Error violating Max limit 0x3 : Error violating Min Limit Reset type: SYSRSn
15-14	SIG2CMPBERR	R/W	0h	This status indicates the error related to measurement (width or frequency) out of bound for SIG2 against Compare B margins 0x0 : No Error 0x1 : Reserved (No Error Default) 0x2 : Error violating Max limit 0x3 : Error violating Min Limit Reset type: SYSRSn
13-12	SIG2AVGERR	R/W	0h	This status indicates the error related to Average Measurement count out of bound for SIG2 aggregation 0x0 : No Error 0x1 : Reserved (No Error Default) 0x2 : Error violating Max limit 0x3 : Error violating Min Limit Reset type: SYSRSn
11-10	SIG2PKERR	R/W	0h	This status indicates the error related to Peak Measurement count out of bound for SIG2 aggregation 0x0 : No Error 0x1 : Reserved (No Error Default) 0x2 : Error violating Max limit 0x3 : Error violating Min Limit Reset type: SYSRSn
9-8	SIG2CMPAERR	R/W	0h	This status indicates the error related to measurement (width or frequency) out of bound for SIG2 against Compare A margins 0x0 : No Error 0x1 : Reserved (No Error Default) 0x2 : Error violating Max limit 0x3 : Error violating Min Limit Reset type: SYSRSn

Table 19-32. BLKERRSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-6	SIG1CMPBERR	R/W	0h	This status indicates the error related to measurement (width or frequency) out of bound for SIG1 against Compare B margins 0x0 : No Error 0x1 : Reserved (No Error Default) 0x2 : Error violating Max limit 0x3 : Error violating Min Limit Reset type: SYSRSn
5-4	SIG1AVGERR	R/W	0h	This status indicates the error related to Average Measurement count out of bound for SIG1 aggregation 0x0 : No Error 0x1 : Reserved (No Error Default) 0x2 : Error violating Max limit 0x3 : Error violating Min Limit Reset type: SYSRSn
3-2	SIG1PKERR	R/W	0h	This status indicates the error related to Peak Measurement count out of bound for SIG1 aggregation 0x0 : No Error 0x1 : Reserved (No Error Default) 0x2 : Error violating Max limit 0x3 : Error violating Min Limit Reset type: SYSRSn
1-0	SIG1CMPAERR	R/W	0h	This status indicates the error related to measurement (width or frequency) out of bound for SIG1 against Compare A margins 0x0: No Error 0x1: Reserved (No Error Default) 0x2: Error violating Max limit 0x3: Error violating Min Limit Reset type: SYSRSn

19.9.2.23 BLKERRINFO Register (Offset = C4h) [Reset = 00000000h]

BLKERRINFO is shown in Figure 19-35 and described in Table 19-33.

Return to the Summary Table.

Block count value of failed register

Figure 19-35. BLKERRINFO Register

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

Table 19-33. BLKERRINFO Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30-28	ERRTYPE	R	0h	This provides the type of error event for which the count is registered following is the encoding for the same. 0x0 : Pulse width Check 0x1 : Pulse width sum Check 0x2 : Edge count Check 0x3 : Moving window edge count Check 0x4 : AVG Check 0x5 : PEAK Check 0x6 : DBAND Check 0x7 : OLAP Check Reset type: SYSRSn
27-25	RESERVED	R	0h	Reserved
24-0	ERRCNT	R	0h	Measurement count of the error is reflected here. The count on last error detection is updated here. Note the count that is selected for read based on ERRCNTSEL of respective BLKERRCFG register. Reset type: SYSRSn

19.9.2.24 BLKERRCFG Register (Offset = C8h) [Reset = 00000000h]

BLKERRCFG is shown in Figure 19-36 and described in Table 19-34.

Return to the Summary Table.

Block Debug additional info.

Figure 19-36. BLKERRCFG Register

31	30	29	28	27	26	25	24
RESERVED				OVERIDESIGTOSIG			
R-0h				R/W-0h			
23	22	21	20	19	18	17	16
RESERVED				OVERIDESIG2			
R-0h				R/W-0h			
15	14	13	12	11	10	9	8
RESERVED				OVERIDESIG1			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED						ERRCNTSEL	
R-0h						R/W-0h	

Table 19-34. BLKERRCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Reserved
27-24	OVERIDESIGTOSIG	R/W	0h	If there is error on SIGTOSIG check take action as per selection 0x0 : No action or override 0x1-0x3 : Reserved 0x4 : Drive the output SIG2OUT, SIG1OUT : '00' 0x5 : Drive the output SIG2OUT, SIG1OUT: '01' 0x6 : Drive the output SIG2OUT, SIG1OUT: '10' 0x7 : Drive the output SIG2OUT, SIG1OUT: '11' 0x8-0xF : Reserved Reset type: SYSRSn
23-20	RESERVED	R	0h	Reserved
19-16	OVERIDESIG2	R/W	0h	If there is error on SIG2 check take action as per selection 0x0 : No action or override 0x1 : Reserved 0x2 : Drive the SIG2 output (SIG2OUT) to '0' 0x3 : Drive the SIG2 output (SIG2OUT) to '1' 0x4-0xF : Reserved Reset type: SYSRSn
15-12	RESERVED	R	0h	Reserved
11-8	OVERIDESIG1	R/W	0h	If there is error on SIG1 check take action as per selection 0x0 : No action or override 0x1 : Reserved 0x2 : Drive the SIG1 output (SIG1 OUT) to '0' 0x3 : Drive the SIG1 output (SIG1 OUT) to '1' 0x4-0xF : Reserved Reset type: SYSRSn
7-2	RESERVED	R	0h	Reserved

Table 19-34. BLKERRCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	ERRCNTSEL	R/W	0h	Depending upon the select value multiplexes the count to BLKERRINFO[ERRCNT] 0x0 : SIG1 Debug count 0x1 : SIG2 Debug count 0x2 : SIGTOSIG Debug count 0x3 : Reserved Reset type: SYSRSn

19.9.2.25 SSS_EVTMASK Register (Offset = CCh) [Reset = 00000000h]

SSS_EVTMASK is shown in Figure 19-37 and described in Table 19-35.

Return to the Summary Table.

Mask for Error word to OR (active high) set of error events to be aggregated to trigger the safe sequence trigger. Output from WADI block

Figure 19-37. SSS_EVTMASK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				SIG2MASK			
R-0h				R/W-0h			
15	14	13	12	11	10	9	8
RESERVED	SIG1MASK						
R-0h	R/W-0h						
7	6	5	4	3	2	1	0
SIG1MASK							
R/W-0h							

Table 19-35. SSS_EVTMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19-16	SIG2MASK	R/W	0h	Each bit can mask the SSS_TRIGEVT to control the aggregation of the cause to Safe State Sequence(SSS) event word bit for SIG2 event or not. Bit Set to 0 then event is not aggregated (ORed) if corresponding bit is set then the event is aggregated. SIG2 events are aggregated on the odd bits of SSS_EVTTRIG[EVTTRIG] [16] : SIG2CMPAERR [17] : SIG2PKERR [18] : SIG2AVGERR [19] : SIG2CMPBERR Reset type: SYSRSn
15	RESERVED	R	0h	Reserved
14-0	SIG1MASK	R/W	0h	Each bit can mask the SSS_TRIGEVT to control the aggregation of the cause to Safe State Sequence(SSS) event word or not. Bit Set to 0 then event is not aggregated (ORed) if corresponding bit is set then the event is aggregated. SIG1 is primary signal in block and also holds mask for SIGTOSIG compares. Both SIG1 and SIGTOSIG are ORed onto even bits of SSS_EVTTRIG[EVTTRIG] [0] : SIG1CMPAERR [1] : SIG1PKERR [2] : SIG1AVGERR [3] : SIG1CMPBERR [7:4] : Reserved , have no effect of SSS event word [8] : SIGTOSIG_CMPERR [9] : SIGTOSIG_PKERR [10] : SIGTOSIG_AVGERR [11] : SIGTOSIG_DBCMPAERR [12] : SIGTOSIG_OLAPCMPAERR [13] : SIGTOSIG_DBCMPBERR [14] : SIGTOSIG_OLAPCMPBERR Reset type: SYSRSn

19.9.2.26 PARTEST Register (Offset = E8h) [Reset = 00000000h]

PARTEST is shown in Figure 19-38 and described in Table 19-36.

Return to the Summary Table.

Enables parity test

Figure 19-38. PARTEST Register

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

Table 19-36. PARTEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	TESTEN	R/W	0h	1010: Parity test feature is enabled All other values: Parity test feature is disabled Note: (1) When the parity test feature is enabled, actual registers are not accessible in the memory map. Instead, the parity values (final XOR output indicating the parity error) are accessible Parity is computed for every byte and the corresponding parity error value is available at the bit-0 of every byte. Value of '1' written to the parity bit after enabling the parity test feature can be used to inject the error by inverting the stored parity value (2) It is recommended to leave the field as 0101 or 0000 after completing the parity test Reset type: SYSRSn

19.9.3 WADI_OPER_SSS_REGS Registers

Table 19-37 lists the memory-mapped registers for the WADI_OPER_SSS_REGS registers. All register offset addresses not listed in Table 19-37 should be considered as reserved locations and the register contents should not be modified.

Table 19-37. WADI_OPER_SSS_REGS Registers

Offset	Acronym	Register Name	Protection
0h	BASETIMERLOW	Read only value for lower word of the base timer.	PARITY_PROTECTED
4h	BASETIMERHIGH	Upper word of the base timer and trigger to start the timer.	
8h	INTSTS	Gives the block wise aggregated raw error status that shall trigger the interrupt.	
Ch	INTSTSMASK	Provides mask for application to suppress or allow a particular block related errors to cause interrupt or not	
10h	BLKSMASKSTS	This is AND combination of active high indication of block-wise status asserting an interrupt.	
14h	INTSTSCLR	Provides way for application to clear the interrupt status.	
18h	INTSTSFRC	Allows emulation of the RIS to raise interrupt in software	
1Ch	SIGSYNCFILTCFG	Each input SIG to be connected through synchroniser or not	PARITY_PROTECTED
20h	TRIGSYNCFILTCFG	Each trigger to be connected through synchroniser or not	
3Ch	REVISION	IP Revision	
40h	DMATRIGSTS	Status of the DMA triggers active	
44h	DMATRIGEN	Mask to enable individual trigger cause	PARITY_PROTECTED
48h	DMASTSUPDATE	Indicates the register set updated post DMA write	
4Ch	DMAFILTWRCFG	DMA writes Filter enable for each of WADI & SSS blocks	PARITY_PROTECTED
A4h	CFGREGLOCK	Configuration Register Lock	
A8h	CFGREGCOMMIT	Configuration Register Commit	
ACh	OPERREGLOCK	Operating Register Lock	
B0h	OPERREGCOMMIT	Operating Register Commit	
B8h	SSS_EVTTRIG	Event trigger word output by WADI instance for SSS	PARITY_PROTECTED
BCh	SSS_OUTEVTSTS	Status of the active event	
C0h	SSS_BLK1_2OUTSEL	Mux select word for outputs 0-3	PARITY_PROTECTED
C4h	SSS_BLK3_4OUTSEL	Mux select word for outputs 4-7	PARITY_PROTECTED
E0h	SSS_OUTEVT1LINKCFG	Output event1 Link configuration	PARITY_PROTECTED
E4h	SSS_OUTEVT2LINKCFG	Output event2 link configuration	PARITY_PROTECTED
E8h	SSS_OUTEVT3LINKCFG	Output event3 link configuration	PARITY_PROTECTED
ECh	SSS_OUTEVT4LINKCFG	Output event4 link configuration	PARITY_PROTECTED
F0h	SSS_OUTEVT5LINKCFG	Output event5 link configuration	PARITY_PROTECTED
F4h	SSS_OUTEVT6LINKCFG	Output event6 link configuration	PARITY_PROTECTED
F8h	SSS_OUTEVT7LINKCFG	Output event7 link configuration	PARITY_PROTECTED

Table 19-37. WADI_OPER_SSS_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
FCh	SSS_OUTEVT8LINKCFG	Output event8 link configuration	PARITY_PROTECTED
100h	SSS_EVT1CFG	Event1 that compares to trigger word SSS_EVTTRIG	PARITY_PROTECTED
104h	SSS_EVT2CFG	Event2 that compares to trigger word SSS_EVTTRIG	PARITY_PROTECTED
108h	SSS_EVT3CFG	Event3 that compares to trigger word SSS_EVTTRIG	PARITY_PROTECTED
10Ch	SSS_EVT4CFG	Event4 that compares to trigger word SSS_EVTTRIG	PARITY_PROTECTED
110h	SSS_TRIGEVT1_4CFG	Configuration for using trigger word in specific manner	PARITY_PROTECTED
180h	SSS_BLKSOUTEVT1CFG	Output event1 to be applied for output	PARITY_PROTECTED
184h	SSS_BLKSOUTEVT2CFG	Output event2 to be applied for output	PARITY_PROTECTED
188h	SSS_BLKSOUTEVT3CFG	Output event3 to be applied for output	PARITY_PROTECTED
18Ch	SSS_BLKSOUTEVT4CFG	Output event4 to be applied for output	PARITY_PROTECTED
190h	SSS_OUTEVT1TRIGCFG	Output event1 configuration for triggers, link	PARITY_PROTECTED
194h	SSS_OUTEVT2TRIGCFG	Output event2 configuration for triggers, link	PARITY_PROTECTED
198h	SSS_OUTEVT3TRIGCFG	Output event3 configuration for triggers, link	PARITY_PROTECTED
19Ch	SSS_OUTEVT4TRIGCFG	Output event4 configuration for triggers, link	PARITY_PROTECTED
1A0h	SSS_OUTEVT1DUR	Output event1 configuration of delays	PARITY_PROTECTED
1A4h	SSS_OUTEVT2DUR	Output event2 configuration of delays	PARITY_PROTECTED
1A8h	SSS_OUTEVT3DUR	Output event3 configuration of delays	PARITY_PROTECTED
1ACh	SSS_OUTEVT4DUR	Output event4 configuration of delays	PARITY_PROTECTED
200h	SSS_EVT5CFG	Event5 that compares to trigger word SSS_EVTTRIG	PARITY_PROTECTED
204h	SSS_EVT6CFG	Event6 that compares to trigger word SSS_EVTTRIG	PARITY_PROTECTED
208h	SSS_EVT7CFG	Event7 that compares to trigger word SSS_EVTTRIG	PARITY_PROTECTED
20Ch	SSS_EVT8CFG	Event8 that compares to trigger word SSS_EVTTRIG	PARITY_PROTECTED
210h	SSS_TRIGEVT5_8CFG	Configuration for using trigger word in specific manner	PARITY_PROTECTED
280h	SSS_BLKSOUTEVT5CFG	Output event5 to be applied for output	PARITY_PROTECTED
284h	SSS_BLKSOUTEVT6CFG	Output event6 to be applied for output	PARITY_PROTECTED
288h	SSS_BLKSOUTEVT7CFG	Output event7 to be applied for output	PARITY_PROTECTED

Table 19-37. WADI_OPER_SSS_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
28Ch	SSS_BLKSOUTEVT8CFG	Output event8 to be applied for output	PARITY_PROTECTED
290h	SSS_OUTEVT5TRIGCFG	Output event5 configuration for triggers, link	PARITY_PROTECTED
294h	SSS_OUTEVT6TRIGCFG	Output event6 configuration for triggers, link	PARITY_PROTECTED
298h	SSS_OUTEVT7TRIGCFG	Output event7 configuration for triggers, link	PARITY_PROTECTED
29Ch	SSS_OUTEVT8TRIGCFG	Output event8 configuration for triggers, link	PARITY_PROTECTED
2A0h	SSS_OUTEVT5DUR	Output event5 configuration of delays	PARITY_PROTECTED
2A4h	SSS_OUTEVT6DUR	Output event6 configuration of delays	PARITY_PROTECTED
2A8h	SSS_OUTEVT7DUR	Output event7 configuration of delays	PARITY_PROTECTED
2ACh	SSS_OUTEVT8DUR	Output event8 configuration of delays	PARITY_PROTECTED
2C8h	PARTEST	Enables parity test	

Complex bit access types are encoded to fit into small table cells. Table 19-38 shows the codes that are used for access types in this section.

Table 19-38. WADI_OPER_SSS_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W0C	W 0C	Write 0 to clear
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

19.9.3.1 BASETIMERLOW Register (Offset = 0h) [Reset = 00000000h]

BASETIMERLOW is shown in Figure 19-39 and described in Table 19-39.

Return to the Summary Table.

Reflects the latest read value of common time base lower word.

Figure 19-39. BASETIMERLOW Register

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

Table 19-39. BASETIMERLOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LOWWORD	R	0h	The lower 32 bits of the base timer for debug Reset type: SYSRSn

19.9.3.2 BASETIMERHIGH Register (Offset = 4h) [Reset = 00000000h]

BASETIMERHIGH is shown in Figure 19-40 and described in Table 19-40.

Return to the Summary Table.

Has latest read value of the common time base of the upper word. Also has time-base start control. This is write once after reset, configured by application after clocks and resets are stable.

Figure 19-40. BASETIMERHIGH Register

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

Table 19-40. BASETIMERHIGH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ENBASETIMER	R/W	0h	When Set this starts the base timer for the WADI block which is commonly used timer for all WADI blocks Reset type: SYSRSn
30-28	RESERVED	R	0h	Reserved
27-0	HIGHWORD	R	0h	The upper 32 bits of the base timer for debug Reset type: SYSRSn

19.9.3.3 INTSTS Register (Offset = 8h) [Reset = 00000000h]

INTSTS is shown in Figure 19-41 and described in Table 19-41.

Return to the Summary Table.

This is 32 bit status register aggregating the error status of each WADI block to trigger an interrupt to CPU.

Figure 19-41. INTSTS Register

31	30	29	28	27	26	25	24
SIGTOSIG_BLKSINT	SIGBLKSINT	RESERVED					
R-0h	R-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	BLK4INT	BLK3INT	BLK2INT	BLK1INT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 19-41. INTSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SIGTOSIG_BLKSINT	R	0h	This register provides the raw information about the presence of the mismatch error between SIGs or availability of readings across WADI blocks as reported through the BLKERRSTS[SIGTOSIG_ERR] All the WADI block SIGTOSIG mismatch errors are aggregated together by ORing Reset type: SYSRSn
30	SIGBLKSINT	R	0h	This register provides the raw information about the presence of the failed waveform measurement due to signal Anomaly as reported through the BLKERRSTS[SIG_ERR] All the WADI block measurement errors are aggregated together by Oring Reset type: SYSRSn
29-16	RESERVED	R	0h	Reserved
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved

Table 19-41. INTSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	BLK4INT	R	0h	This register provides the raw information about the presence of the error per WADI block as reported through the BLKERRSTS The Oared version of each block reflects index-wise in each bit of this register Reset type: SYSRSn
2	BLK3INT	R	0h	This register provides the raw information about the presence of the error per WADI block as reported through the BLKERRSTS The Oared version of each block reflects index-wise in each bit of this register Reset type: SYSRSn
1	BLK2INT	R	0h	This register provides the raw information about the presence of the error per WADI block as reported through the BLKERRSTS The Oared version of each block reflects index-wise in each bit of this register Reset type: SYSRSn
0	BLK1INT	R	0h	This register provides the raw information about the presence of the error per WADI block as reported through the BLKERRSTS The Oared version of each block reflects index-wise in each bit of this register Reset type: SYSRSn

19.9.3.4 INTSTSMASK Register (Offset = Ch) [Reset = 00000000h]

INTSTSMASK is shown in Figure 19-42 and described in Table 19-42.

Return to the Summary Table.

This is mask register to suppress the errors related to specific WADI block.

Figure 19-42. INTSTSMASK Register

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

Table 19-42. INTSTSMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SIGTOSIG_BLKSMASK	R/W	0h	This bit is representative of the masking of mismatch error aggregated across WADI blocks. '0': corresponding interrupt is not asserted to the CPU. '1': related interrupt is allowed to cause an interrupt to CPU. Reset type: SYSRSn
30	SIGBLKSMASK	R/W	0h	This bit is representative of the masking of measurement error aggregated across WADI blocks. '0': corresponding interrupt is not asserted to the CPU. '1': related interrupt is allowed to cause an interrupt to CPU. Reset type: SYSRSn
29-16	RESERVED	R	0h	Reserved
15-0	BLKSMASK	R/W	0h	Each register bit is representative of the block and can enable or mask the corresponding error to trigger an interrupt or not to the CPU '0': corresponding interrupt is not asserted to the CPU '1': related interrupt is allowed to cause an interrupt to CPU Reset type: SYSRSn

19.9.3.5 BLKSMASKSTS Register (Offset = 10h) [Reset = 00000000h]

BLKSMASKSTS is shown in Figure 19-43 and described in Table 19-43.

Return to the Summary Table.

This is masked interrupt status for the WADI IP that triggered the interrupt to CPU. ISR can directly read this to check which WADI block(s) caused an interrupt.

Figure 19-43. BLKSMASKSTS Register

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

Table 19-43. BLKSMASKSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SIGTOSIGBLKS	R	0h	This register gives the bit wise status of activated and asserted error bits block-wise Hence interrupt service routine can readily reach to the detailed status of respective block Reset type: SYSRSn
30	SIGBLKS	R	0h	This register gives the bit wise status of activated and asserted error bits block-wise Hence interrupt service routine can readily reach to the detailed status of respective block Reset type: SYSRSn
29-16	RESERVED	R	0h	Reserved
15-0	BLKSMASK	R	0h	This register gives the bit wise status of activated and asserted error bits block-wise Hence interrupt service routine can readily reach to the detailed status of respective block Reset type: SYSRSn

19.9.3.6 INTSTSCLR Register (Offset = 14h) [Reset = 00000000h]

INTSTSCLR is shown in Figure 19-44 and described in Table 19-44.

Return to the Summary Table.

This is for interrupt service routines to clear the aggregated status of the block error as in RAW_INT_STATUS.

Figure 19-44. INTSTSCLR Register

31	30	29	28	27	26	25	24
SIGTOSIG_BLKSINT	SIGBLKSINT	RESERVED					
R-0/W1C-0h	R-0/W1C-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	BLK4INT	BLK3INT	BLK2INT	BLK1INT
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 19-44. INTSTSCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SIGTOSIG_BLKSINT	R-0/W1C	0h	This write 1 to clear register clears the INTSTS of corresponding bit that is written 1 This is regardless of the INTSTSMASK Write 0 has no effect Register always reads 0x0 Reset type: SYSRSn
30	SIGBLKSINT	R-0/W1C	0h	This write 1 to clear register clears the INTSTS of corresponding bit that is written 1 This is regardless of the INTSTSMASK Write 0 has no effect Register always reads 0x0 Reset type: SYSRSn
29-16	RESERVED	R	0h	Reserved
15	RESERVED	R-0/W1C	0h	Reserved
14	RESERVED	R-0/W1C	0h	Reserved
13	RESERVED	R-0/W1C	0h	Reserved
12	RESERVED	R-0/W1C	0h	Reserved
11	RESERVED	R-0/W1C	0h	Reserved
10	RESERVED	R-0/W1C	0h	Reserved
9	RESERVED	R-0/W1C	0h	Reserved
8	RESERVED	R-0/W1C	0h	Reserved
7	RESERVED	R-0/W1C	0h	Reserved
6	RESERVED	R-0/W1C	0h	Reserved
5	RESERVED	R-0/W1C	0h	Reserved
4	RESERVED	R-0/W1C	0h	Reserved

Table 19-44. INTSTSCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	BLK4INT	R-0/W1C	0h	This write 1 to clear register clears the INTSTS of corresponding bit that is written 1 This is regardless of the INTSTSMASK Write 0 has no effect Register always reads 0x0 Reset type: SYSRSn
2	BLK3INT	R-0/W1C	0h	This write 1 to clear register clears the INTSTS of corresponding bit that is written 1 This is regardless of the INTSTSMASK Write 0 has no effect Register always reads 0x0 Reset type: SYSRSn
1	BLK2INT	R-0/W1C	0h	This write 1 to clear register clears the INTSTS of corresponding bit that is written 1 This is regardless of the INTSTSMASK Write 0 has no effect Register always reads 0x0 Reset type: SYSRSn
0	BLK1INT	R-0/W1C	0h	This write 1 to clear register clears the INTSTS of corresponding bit that is written 1 This is regardless of the INTSTSMASK Write 0 has no effect Register always reads 0x0 Reset type: SYSRSn

19.9.3.7 INTSTSFRC Register (Offset = 18h) [Reset = 00000000h]

INTSTSFRC is shown in Figure 19-45 and described in Table 19-45.

Return to the Summary Table.

This is 32 bit raw interrupt status emulate register can be used for test and diagnostic of WADI interrupt mechanism.

Figure 19-45. INTSTSFRC Register

31	30	29	28	27	26	25	24
SIGTOSIG_BLKSINT	SIGBLKSINT	RESERVED					
R-0/W1S-0h	R-0/W1S-0h	R-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	BLK4INT	BLK3INT	BLK2INT	BLK1INT
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 19-45. INTSTSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SIGTOSIG_BLKSINT	R-0/W1S	0h	This interrupt status emulate field can be used for test and diagnose the WADI interrupt mechanism. Write 1 forces the corresponding Interrupt bit. Write 0 has no effect. Reset type: SYSRSn
30	SIGBLKSINT	R-0/W1S	0h	This interrupt status emulate field can be used for test and diagnose the WADI interrupt mechanism. Write 1 forces the corresponding Interrupt bit. Write 0 has no effect. Reset type: SYSRSn
29-16	RESERVED	R	0h	Reserved
15	RESERVED	R-0/W1S	0h	Reserved
14	RESERVED	R-0/W1S	0h	Reserved
13	RESERVED	R-0/W1S	0h	Reserved
12	RESERVED	R-0/W1S	0h	Reserved
11	RESERVED	R-0/W1S	0h	Reserved
10	RESERVED	R-0/W1S	0h	Reserved
9	RESERVED	R-0/W1S	0h	Reserved
8	RESERVED	R-0/W1S	0h	Reserved
7	RESERVED	R-0/W1S	0h	Reserved
6	RESERVED	R-0/W1S	0h	Reserved
5	RESERVED	R-0/W1S	0h	Reserved
4	RESERVED	R-0/W1S	0h	Reserved
3	BLK4INT	R-0/W1S	0h	This interrupt status emulate field can be used for test and diagnose the WADI interrupt mechanism. Write 1 forces the corresponding Interrupt bit. Write 0 has no effect. Reset type: SYSRSn

Table 19-45. INTSTSFRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	BLK3INT	R-0/W1S	0h	This interrupt status emulate field can be used for test and diagnose the WADI interrupt mechanism. Write 1 forces the corresponding Interrupt bit. Write 0 has no effect. Reset type: SYSRSn
1	BLK2INT	R-0/W1S	0h	This interrupt status emulate field can be used for test and diagnose the WADI interrupt mechanism. Write 1 forces the corresponding Interrupt bit. Write 0 has no effect. Reset type: SYSRSn
0	BLK1INT	R-0/W1S	0h	This interrupt status emulate field can be used for test and diagnose the WADI interrupt mechanism. Write 1 forces the corresponding Interrupt bit. Write 0 has no effect. Reset type: SYSRSn

19.9.3.8 SIGSYNCFILTCFG Register (Offset = 1Ch) [Reset = 00000000h]

SIGSYNCFILTCFG is shown in Figure 19-46 and described in Table 19-46.

Return to the Summary Table.

This is 32 bit mux select to either route the raw input signal to WADI block or to connect through synchroniser and filter.

Figure 19-46. SIGSYNCFILTCFG Register

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

Table 19-46. SIGSYNCFILTCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	Reserved
7-0	BLKS_SIGS_SYNCFILT	R/W	0h	This word provides bitwise option to select the mused input to WADI block connected from cross-bar either directly (1'b0) or through synchroniser followed by glitch filter for width of 1 clock. The resultant signal when elected through sync & filter is delayed by as many (3-4) clock cycles. [0] : BLK1.SIG1 Connected through synchroniser and glitch filter [1] : BLK1.SIG2 Connected through synchroniser and glitch filter [2] : BLK2.SIG1 Connected through synchroniser and glitch filter [3] : BLK2.SIG2 Connected through synchroniser and glitch filter [4] : BLK3.SIG1 Connected through synchroniser and glitch filter [5] : BLK3.SIG2 Connected through synchroniser and glitch filter [6] : BLK4.SIG1 Connected through synchroniser and glitch filter [7] : BLK4.SIG2 Connected through synchroniser and glitch filter Reset type: SYSRSn

19.9.3.9 TRIGSYNCFILTCFG Register (Offset = 20h) [Reset = 00000000h]

TRIGSYNCFILTCFG is shown in Figure 19-47 and described in Table 19-47.

Return to the Summary Table.

This is 32 bit mux select to either route the trigger signal directly to WADI block or to connect through synchroniser and filter.

Figure 19-47. TRIGSYNCFILTCFG Register

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

Table 19-47. TRIGSYNCFILTCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	Reserved
7-0	BLKS_TRIG_SYNCFILT	R/W	0h	This word provides bitwise option to select the muxed trigger input to WADI block connected from cross-bar either directly (1'b0) or through synchroniser followed by glitch filter for width of 1 clock. The resultant signal when elected through sync & filter is delayed by as many (3-4) clock cycles. [0] : TRIG input for BLK1.SIG1 Connected through synchroniser and glitch filter [1] : TRIG input for BLK1.SIG2 Connected through synchroniser and glitch filter [2] : TRIG input for BLK2.SIG1 Connected through synchroniser and glitch filter [3] : TRIG input for BLK2.SIG2 Connected through synchroniser and glitch filter [4] : TRIG input for BLK3.SIG1 Connected through synchroniser and glitch filter [5] : TRIG input for BLK3.SIG2 Connected through synchroniser and glitch filter [6] : TRIG input for BLK4.SIG1 Connected through synchroniser and glitch filter [7] : TRIG input for BLK4.SIG2 Connected through synchroniser and glitch filter Reset type: SYSRSn

19.9.3.10 REVISION Register (Offset = 3Ch) [Reset = 4FF00100h]

REVISION is shown in Figure 19-48 and described in Table 19-48.

Return to the Summary Table.

IP Revision

Figure 19-48. REVISION Register

31	30	29	28	27	26	25	24
SCHEME		RESERVED		FUNC			
R-1h		R-0-0h		R-FF0h			
23	22	21	20	19	18	17	16
FUNC							
R-FF0h							
15	14	13	12	11	10	9	8
RTL					MAJOR		
R-0h					R-1h		
7	6	5	4	3	2	1	0
CUSTOM		MINOR					
R-0h		R-0h					

Table 19-48. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	This identifies the scheme revision ID register type implemented for this module Reset type: SYSRSn
29-28	RESERVED	R-0	0h	Reserved
27-16	FUNC	R	FF0h	Functional Release Number Reflects software-compatability. If there is no software compatability, a unique func number is assigned for compatible modules, the same number is maintained Reset type: SYSRSn
15-11	RTL	R	0h	Design Release Number Incremented for releases due to spec changes or post-release design changes Reset to 0 when either MAJOR or MINOR is incremented Reset type: SYSRSn
10-8	MAJOR	R	1h	Major Revision Number Represents major changes to the module (e.g. entirely new features are added/changed) The major revision number for this module Reset type: SYSRSn
7-6	CUSTOM	R	0h	Custom Module Number Indicates a special version of the module May not be supported by standard software Reset type: SYSRSn
5-0	MINOR	R	0h	Minor Revision Number Represents minor changes to the module (e.g. enhancements to existing features) The minor revision number for this module Reset type: SYSRSn

19.9.3.11 DMATRIGSTS Register (Offset = 40h) [Reset = 00000000h]

DMATRIGSTS is shown in Figure 19-49 and described in Table 19-49.

Return to the Summary Table.

This 32b DMA status indicate the WADI Block/SIG that triggered DMA request

Figure 19-49. DMATRIGSTS Register

31	30	29	28	27	26	25	24
SSS_OUTEVT8	SSS_OUTEVT7	SSS_OUTEVT6	SSS_OUTEVT5	SSS_OUTEVT4	SSS_OUTEVT3	SSS_OUTEVT2	SSS_OUTEVT1
R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h
23	22	21	20	19	18	17	16
SSS_EVT8	SSS_EVT7	SSS_EVT6	SSS_EVT5	SSS_EVT4	SSS_EVT3	SSS_EVT2	SSS_EVT1
R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	BLK4_DMAEVT	BLK3_DMAEVT	BLK2_DMAEVT	BLK1_DMAEVT
R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h

Table 19-49. DMATRIGSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SSS_OUTEVT8	R/W0C	0h	Each bit is indicative of the SSS Event word event that triggered the DMA event. The status is taken after applying the DMA_Event_Enable mask. Following are the assignments for different Sequencer events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of OUTEVT8 triggered DMA Request Reset type: SYSRSn
30	SSS_OUTEVT7	R/W0C	0h	Each bit is indicative of the SSS Event word event that triggered the DMA event. The status is taken after applying the DMA_Event_Enable mask. Following are the assignments for different Sequencer events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of OUTEVT7 triggered DMA Request Reset type: SYSRSn
29	SSS_OUTEVT6	R/W0C	0h	Each bit is indicative of the SSS Event word event that triggered the DMA event. The status is taken after applying the DMA_Event_Enable mask. Following are the assignments for different Sequencer events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of OUTEVT6 triggered DMA Request Reset type: SYSRSn

Table 19-49. DMATRIGSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	SSS_OUTEVT5	R/W0C	0h	Each bit is indicative of the SSS Event word event that triggered the DMA event. The status is taken after applying the DMA_Event_Enable mask. Following are the assignments for different Sequencer events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of OUTEVT5 triggered DMA Request Reset type: SYSRSn
27	SSS_OUTEVT4	R/W0C	0h	Each bit is indicative of the SSS Event word event that triggered the DMA event. The status is taken after applying the DMA_Event_Enable mask. Following are the assignments for different Sequencer events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of OUTEVT4 triggered DMA Request Reset type: SYSRSn
26	SSS_OUTEVT3	R/W0C	0h	Each bit is indicative of the SSS Event word event that triggered the DMA event. The status is taken after applying the DMA_Event_Enable mask. Following are the assignments for different Sequencer events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of OUTEVT3 triggered DMA Request Reset type: SYSRSn
25	SSS_OUTEVT2	R/W0C	0h	Each bit is indicative of the SSS Event word event that triggered the DMA event. The status is taken after applying the DMA_Event_Enable mask. Following are the assignments for different Sequencer events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of OUTEVT2 triggered DMA Request Reset type: SYSRSn
24	SSS_OUTEVT1	R/W0C	0h	Each bit is indicative of the SSS Event word event that triggered the DMA event. The status is taken after applying the DMA_Event_Enable mask. Following are the assignments for different Sequencer events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of OUTEVT1 triggered DMA Request Reset type: SYSRSn
23	SSS_EVT8	R/W0C	0h	Each bit is indicative of the SSS Event word event that triggered the DMA event. The status is taken after applying the DMA_Event_Enable mask. Following are the assignments for different events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of EVENT_WORD8 triggered DMA Request Reset type: SYSRSn

Table 19-49. DMATRIGSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	SSS_EVT7	R/W0C	0h	Each bit is indicative of the SSS Event word event that triggered the DMA event. The status is taken after applying the DMA_Event_Enable mask. Following are the assignments for different events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of EVENT_WORD7 triggered DMA Request Reset type: SYSRSn
21	SSS_EVT6	R/W0C	0h	Each bit is indicative of the SSS Event word event that triggered the DMA event. The status is taken after applying the DMA_Event_Enable mask. Following are the assignments for different events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of EVENT_WORD6 triggered DMA Request Reset type: SYSRSn
20	SSS_EVT5	R/W0C	0h	Each bit is indicative of the SSS Event word event that triggered the DMA event. The status is taken after applying the DMA_Event_Enable mask. Following are the assignments for different events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of EVENT_WORD5 triggered DMA Request Reset type: SYSRSn
19	SSS_EVT4	R/W0C	0h	Each bit is indicative of the SSS Event word event that triggered the DMA event. The status is taken after applying the DMA_Event_Enable mask. Following are the assignments for different events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of EVENT_WORD4 triggered DMA Request Reset type: SYSRSn
18	SSS_EVT3	R/W0C	0h	Each bit is indicative of the SSS Event word event that triggered the DMA event. The status is taken after applying the DMA_Event_Enable mask. Following are the assignments for different events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of EVENT_WORD3 triggered DMA Request Reset type: SYSRSn
17	SSS_EVT2	R/W0C	0h	Each bit is indicative of the SSS Event word event that triggered the DMA event. The status is taken after applying the DMA_Event_Enable mask. Following are the assignments for different events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of EVENT_WORD2 triggered DMA Request Reset type: SYSRSn

Table 19-49. DMATRIGSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	SSS_EVT1	R/W0C	0h	Each bit is indicative of the SSS Event word event that triggered the DMA event. The status is taken after applying the DMA_Event_Enable mask. Following are the assignments for different events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of EVENT_WORD1 triggered DMA Request Reset type: SYSRSn
15	RESERVED	R/W0C	0h	Reserved
14	RESERVED	R/W0C	0h	Reserved
13	RESERVED	R/W0C	0h	Reserved
12	RESERVED	R/W0C	0h	Reserved
11	RESERVED	R/W0C	0h	Reserved
10	RESERVED	R/W0C	0h	Reserved
9	RESERVED	R/W0C	0h	Reserved
8	RESERVED	R/W0C	0h	Reserved
7	RESERVED	R/W0C	0h	Reserved
6	RESERVED	R/W0C	0h	Reserved
5	RESERVED	R/W0C	0h	Reserved
4	RESERVED	R/W0C	0h	Reserved
3	BLK4_DMAEVT	R/W0C	0h	Each bit is indicative of the WADI event that triggered the DMA event. The status is taken after applying the mask. Following are the assignments for different events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of block3 that triggered DMA Request Reset type: SYSRSn
2	BLK3_DMAEVT	R/W0C	0h	Each bit is indicative of the WADI event that triggered the DMA event. The status is taken after applying the mask. Following are the assignments for different events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of block2 that triggered DMA Request Reset type: SYSRSn
1	BLK2_DMAEVT	R/W0C	0h	Each bit is indicative of the WADI event that triggered the DMA event. The status is taken after applying the mask. Following are the assignments for different events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of block1 that triggered DMA Request Reset type: SYSRSn

Table 19-49. DMATRIGSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	BLK1_DMAEVT	R/W0C	0h	Each bit is indicative of the WADI event that triggered the DMA event. The status is taken after applying the mask. Following are the assignments for different events. Status bit when set indicates occurrence of event. Upon DMA write and DMA ack the bit will get cleared. Alternatively Write of 0x0 by user shall also clear the corresponding bit status. Write of 0x1 has no effect on bit. DMA enabled event of block0 that triggered DMA Request Reset type: SYSRSn

19.9.3.12 DMATRIGEN Register (Offset = 44h) [Reset = 00000000h]

DMATRIGEN is shown in Figure 19-50 and described in Table 19-50.

Return to the Summary Table.

This 32b control to individually enable DMA trigger reasons of different Block SIG errors

Figure 19-50. DMATRIGEN Register

31	30	29	28	27	26	25	24
SSS_OUTEVT5_8DMA				SSS_OUTEVT1_4DMA			
R/W-0h				R/W-0h			
23	22	21	20	19	18	17	16
SSS_EVT5_8DMA				SSS_EVT1_4DMA			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
BLKSDMAEVT							
R/W-0h							
7	6	5	4	3	2	1	0
BLKSDMAEVT							
R/W-0h							

Table 19-50. DMATRIGEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	SSS_OUTEVT5_8DMA	R/W	0h	Application needs to enable respective bits to ensure the DMA request is raised based on the event. 0x1 configuration enables DMA request assertion, 0x0 is disable for the event and does not cause the DMA request. Each bit is indicative of the SSS output events as below. [28] :DMA Request Enable for event of OUTEVT5 [29] :DMA Request Enable for event of OUTEVT6 [30] :DMA Request Enable for event of OUTEVT7 [31] :DMA Request Enable for event of OUTEVT8 Reset type: SYSRSn
27-24	SSS_OUTEVT1_4DMA	R/W	0h	Application needs to enable respective bits to ensure the DMA request is raised based on the event. 0x1 configuration enables DMA request assertion, 0x0 is disable for the event and does not cause the DMA request. Each bit is indicative of the SSS output events as below. [24] :DMA Request Enable for event of OUTEVT1 [25] :DMA Request Enable for event of OUTEVT2 [26] :DMA Request Enable for event of OUTEVT3 [27] :DMA Request Enable for event of OUTEVT4 Reset type: SYSRSn
23-20	SSS_EVT5_8DMA	R/W	0h	Application needs to enable respective bits to ensure the DMA request is raised based on the event. 0x1 configuration enables DMA request assertion, 0x0 is disable for the event and does not cause the DMA request. Each bit is indicative of the SSS Event word as below. [20] :DMA Request Enable for event of SSS_EVT5 [21] :DMA Request Enable for event of SSS_EVT6 [22] :DMA Request Enable for event of SSS_EVT7 [23] :DMA Request Enable for event of SSS_EVT8 Reset type: SYSRSn

Table 19-50. DMATRIGEN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-16	SSS_EVT1_4DMA	R/W	0h	Application needs to enable respective bits to ensure the DMA request is raised based on the event. 0x1 configuration enables DMA request assertion, 0x0 is disable for the event and does not cause the DMA request. Each bit is indicative of the SSS Event word as below. [16] :DMA Request Enable for event of SSS_EVT1 [17] :DMA Request Enable for event of SSS_EVT2 [18] :DMA Request Enable for event of SSS_EVT3 [19] :DMA Request Enable for event of SSS_EVT4 Reset type: SYSRSn
15-0	BLKSDMAEVT	R/W	0h	Application needs to enable respective bits to ensure the DMA request is raised based on the event. 0x1 configuration enables DMA request assertion, 0x0 is disable for the event and does not cause the DMA request. Each bit is indicative of the WADI Block event as below. [0] :DMA Request Enable for event of block1 [1] :DMA Request Enable for event of block2 [2] :DMA Request Enable for event of block3 [3] :DMA Request Enable for event of block4 [4:15] : Reserved Reset type: SYSRSn

19.9.3.13 DMASTSUPDATE Register (Offset = 48h) [Reset = 00000000h]

DMASTSUPDATE is shown in Figure 19-51 and described in Table 19-51.

Return to the Summary Table.

Status register indicating which register thresholds got updated after the DMA write

Figure 19-51. DMASTSUPDATE Register

31	30	29	28	27	26	25	24
WRSTSOUT8	WRSTSOUT7	WRSTSOUT6	WRSTSOUT5	WRSTSOUT4	WRSTSOUT3	WRSTSOUT2	WRSTSOUT1
R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h
23	22	21	20	19	18	17	16
WRSTSEVT8	WRSTSEVT7	WRSTSEVT6	WRSTSEVT5	WRSTSEVT4	WRSTSEVT3	WRSTSEVT2	WRSTSEVT1
R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	WRSTSBLK4	WRSTSBLK3	WRSTSBLK2	WRSTSBLK1
R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h

Table 19-51. DMASTSUPDATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	WRSTSOUT8	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered DMA transfer or not. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for OUTEVT8 Reset type: SYSRSn
30	WRSTSOUT7	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered DMA transfer or not. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for OUTEVT7 Reset type: SYSRSn
29	WRSTSOUT6	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered DMA transfer or not. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for OUTEVT6 Reset type: SYSRSn
28	WRSTSOUT5	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered DMA transfer or not. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for OUTEVT5 Reset type: SYSRSn
27	WRSTSOUT4	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered DMA transfer or not. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for OUTEVT4 Reset type: SYSRSn
26	WRSTSOUT3	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered DMA transfer or not. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for OUTEVT3 Reset type: SYSRSn

Table 19-51. DMASTSUPDATE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	WRSTSOUT2	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered DMA transfer or not. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for OUTEVT2 Reset type: SYSRSn
24	WRSTSOUT1	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered DMA transfer or not. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for OUTEVT1 Reset type: SYSRSn
23	WRSTSEVT8	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered DMA transfer or not. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for SSS_EVT8 Reset type: SYSRSn
22	WRSTSEVT7	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered DMA transfer or not. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for SSS_EVT7 Reset type: SYSRSn
21	WRSTSEVT6	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered DMA transfer or not. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for SSS_EVT6 Reset type: SYSRSn
20	WRSTSEVT5	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered DMA transfer or not. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for SSS_EVT5 Reset type: SYSRSn
19	WRSTSEVT4	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered DMA transfer or not. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for SSS_EVT4 Reset type: SYSRSn
18	WRSTSEVT3	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered DMA transfer or not. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for SSS_EVT3 Reset type: SYSRSn
17	WRSTSEVT2	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered DMA transfer or not. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for SSS_EVT2 Reset type: SYSRSn
16	WRSTSEVT1	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered DMA transfer or not. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for SSS_EVT1 Reset type: SYSRSn
15	RESERVED	R/W0C	0h	Reserved
14	RESERVED	R/W0C	0h	Reserved
13	RESERVED	R/W0C	0h	Reserved
12	RESERVED	R/W0C	0h	Reserved
11	RESERVED	R/W0C	0h	Reserved
10	RESERVED	R/W0C	0h	Reserved

Table 19-51. DMASTSUPDATE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	RESERVED	R/W0C	0h	Reserved
8	RESERVED	R/W0C	0h	Reserved
7	RESERVED	R/W0C	0h	Reserved
6	RESERVED	R/W0C	0h	Reserved
5	RESERVED	R/W0C	0h	Reserved
4	RESERVED	R/W0C	0h	Reserved
3	WRSTSBLK4	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered transfer or not. The field is not updated if the block is not enabled for filtered write. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for block3 Reset type: SYSRSn
2	WRSTSBLK3	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered transfer or not. The field is not updated if the block is not enabled for filtered write. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for block2 Reset type: SYSRSn
1	WRSTSBLK2	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered transfer or not. The field is not updated if the block is not enabled for filtered write. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for block1 Reset type: SYSRSn
0	WRSTSBLK1	R/W0C	0h	Indicates if MMR configuration of the WADI block is updated by the filtered transfer or not. The field is not updated if the block is not enabled for filtered write. Writing 1 to this register has no effect, Writing 0 clears the staus. Register sets were updated for block0 Reset type: SYSRSn

19.9.3.14 DMAFILTWRCFG Register (Offset = 4Ch) [Reset = 00000000h]

DMAFILTWRCFG is shown in Figure 19-52 and described in Table 19-52.

Return to the Summary Table.

Control for filtered DMA writes to WADI and SSS blocks

Figure 19-52. DMAFILTWRCFG Register

31	30	29	28	27	26	25	24
OUTEVT5_8WREN				OUTEVT1_4WREN			
R/W-0h				R/W-0h			
23	22	21	20	19	18	17	16
EVT5_8WREN				EVT1_4WREN			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
BLKSWREN							
R/W-0h							
7	6	5	4	3	2	1	0
BLKSWREN							
R/W-0h							

Table 19-52. DMAFILTWRCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	OUTEVT5_8WREN	R/W	0h	Enables triggered event cause based DMA writes to the MMR of the Block. When Set the DMA writes have effect only if the event has triggered (post enable) the DMA request. If in cleared (0x0) state then all writes are passed through to the block without filter. [28] :Filtered DMA write enable for OUTEVT5 [29] :Filtered DMA write enable for OUTEVT6 [30] :Filtered DMA write enable for OUTEVT7 [31] :Filtered DMA write enable for OUTEVT8 Reset type: SYSRSn
27-24	OUTEVT1_4WREN	R/W	0h	Enables triggered event cause based DMA writes to the MMR of the Block. When Set the DMA writes have effect only if the event has triggered (post enable) the DMA request. If in cleared (0x0) state then all writes are passed through to the block without filter. [24] :Filtered DMA write enable for OUTEVT1 [25] :Filtered DMA write enable for OUTEVT2 [26] :Filtered DMA write enable for OUTEVT3 [27] :Filtered DMA write enable for OUTEVT4 Reset type: SYSRSn
23-20	EVT5_8WREN	R/W	0h	Enables triggered event cause based DMA writes to the MMR of the Block. When Set the DMA writes have effect only if the event has triggered (post enable) the DMA request. If in cleared (0x0) state then all writes are passed through to the block without filter. [20] :Filtered DMA write enable for SSS_EVT5 [21] :Filtered DMA write enable for SSS_EVT6 [22] :Filtered DMA write enable for SSS_EVT7 [23] :Filtered DMA write enable for SSS_EVT8 Reset type: SYSRSn

Table 19-52. DMAFILTWRCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-16	EVT1_4WREN	R/W	0h	Enables triggered event cause based DMA writes to the MMR of the Block. When Set the DMA writes have effect only if the event has triggered (post enable) the DMA request. If in cleared (0x0) state then all writes are passed through to the block without filter. [16] :Filtered DMA write enable for SSS_EVT1 [17] :Filtered DMA write enable for SSS_EVT2 [18] :Filtered DMA write enable for SSS_EVT3 [19] :Filtered DMA write enable for SSS_EVT4 Reset type: SYSRSn
15-0	BLKSWREN	R/W	0h	Enables triggered event cause based DMA writes to the MMR of the Block. When Set the DMA writes have effect only if the event has triggered (post enable) the DMA request. If in cleared (0x0) state then all writes are passed through to the block without filter. [0] :Filtered DMA write enable for block0 [1] :Filtered DMA write enable for block1 [2] :Filtered DMA write enable for block2 [3] :Filtered DMA write enable for block3 [4] :Filtered DMA write enable for block4 [4:15] : Reserved Reset type: SYSRSn

19.9.3.15 CFGREGLOCK Register (Offset = A4h) [Reset = 00000000h]

CFGREGLOCK is shown in Figure 19-53 and described in Table 19-53.

Return to the Summary Table.

Configuration register lock

Figure 19-53. CFGREGLOCK Register

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

Table 19-53. CFGREGLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	Determines whether WADI Configuration registers can be written. 0 : Register can be written 1 : Register cannot be written This bit can only be modified if CONFIG_REG_COMMIT.COMMIT is zero. Following block MMR are exmpted from lock, due to regular use in opetration. COMMON_CONFIG COMMON_TRIG_CFG ERR_STATUS PARITY_TEST Reset type: SYSRSn

19.9.3.16 CFGREGCOMMIT Register (Offset = A8h) [Reset = 00000000h]

CFGREGCOMMIT is shown in Figure 19-54 and described in Table 19-54.

Return to the Summary Table.

Configuration Register Commit

Figure 19-54. CFGREGCOMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 19-54. CFGREGCOMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the CONFIG_REG_LOCK register. This bit cannot be cleared, except by reset. 0 : CONFIG_REG_LOCK is modifiable 1 : CONFIG_REG_LOCK is committed permanently Reset type: SYSRSn

19.9.3.17 OPERREGLOCK Register (Offset = ACh) [Reset = 00000000h]

OPERREGLOCK is shown in Figure 19-55 and described in Table 19-55.

Return to the Summary Table.

Operating Register Lock

Figure 19-55. OPERREGLOCK Register

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

Table 19-55. OPERREGLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	Determines whether WADI Operational registers can be written. 0 : Register can be written 1 : Register cannot be written This bit can only be modified if OPER_REG_COMMIT.COMMIT is zero. Following block MMR are exmpted from lock, due to regular use in opetration. INT_STATUS_MASK RAW_INT_STATUS_CLR RAW_INT_STATUS_FRC CONFIG_REG_LOCK CONFIG_REG_COMMIT OPER_REG_LOCK OPER_REG_COMMIT PARITY_TEST Reset type: SYSRSn

19.9.3.18 OPERREGCOMMIT Register (Offset = B0h) [Reset = 00000000h]

OPERREGCOMMIT is shown in Figure 19-56 and described in Table 19-56.

Return to the Summary Table.

Operating Register Commit

Figure 19-56. OPERREGCOMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							COMMIT
R-0h							R/W1S-0h

Table 19-56. OPERREGCOMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	COMMIT	R/W1S	0h	When set, locks the OPER_REG_LOCK register. This bit cannot be cleared, except by reset. 0 : OPER_REG_LOCK is modifiable 1 : OPER_REG_LOCK is committed permanently Reset type: SYSRSn

19.9.3.19 SSS_EVTTRIG Register (Offset = B8h) [Reset = 00000000h]

SSS_EVTTRIG is shown in Figure 19-57 and described in Table 19-57.

Return to the Summary Table.

Event trigger word output by WADI instance for SSS

Figure 19-57. SSS_EVTTRIG Register

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

Table 19-57. SSS_EVTTRIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	Reserved
7-0	EVTTRIG	R/W	0h	Each bit is the event recording of a particular signal of the WADI blocks. All odd bits are used for SIG1 and SIGTOSIG events aggregation and odd bits are used for SIG2 event aggregation. The write override to this register is only to enable the emulation/ debug for the event word. [0] : SIG1 & SIGTOSIG event aggregation of block1 [1] : SIG2 event aggregation of block1 [2] : SIG1 & SIGTOSIG event aggregation of block2 [3] : SIG2 event aggregation of block2 [4] : SIG1 & SIGTOSIG event aggregation of block3 [5] : SIG2 event aggregation of block3 [6] : SIG1 & SIGTOSIG event aggregation of block4 [7] : SIG2 event aggregation of block4 Reset type: SYSRSn

19.9.3.20 SSS_OUTEVTSTS Register (Offset = BCh) [Reset = 00000000h]

SSS_OUTEVTSTS is shown in Figure 19-58 and described in Table 19-58.

Return to the Summary Table.

Indicates the active status of output events. Allows emulation by of active output event by overriding the value.

Figure 19-58. SSS_OUTEVTSTS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OUTEVT5_8				OUTEVT1_4			
R-0h								R/W-0h				R/W-0h			

Table 19-58. SSS_OUTEVTSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	RESERVED	R	0h	Reserved
7-4	OUTEVT5_8	R/W	0h	This is status register indicating the active status of the output events as triggered by underlying trigger configurations. Write is allowed to emulate these words to help with debug and development. Setting bit has same effect as output events getting activated in event triggered fashion. Deactivating same as completion of active period but without effect of activation of linked word. [4] :Active Status of OUTEVT5 [5] :Active Status of OUTEVT6 [6] :Active Status of OUTEVT7 [7] :Active Status of OUTEVT8 Reset type: SYSRSn
3-0	OUTEVT1_4	R/W	0h	This is status register indicating the active status of the output events as triggered by underlying trigger configurations. Write is allowed to emulate these words to help with debug and development. Setting bit has same effect as output events getting activated in event triggered fashion. Deactivating same as completion of active period but without effect of activation of linked word. [0] :Active Status of OUTEVT1 [1] :Active Status of OUTEVT2 [2] :Active Status of OUTEVT3 [3] :Active Status of OUTEVT4 Reset type: SYSRSn

19.9.3.21 SSS_BLK1_2OUTSEL Register (Offset = C0h) [Reset = 00000000h]

SSS_BLK1_2OUTSEL is shown in Figure 19-59 and described in Table 19-59.

Return to the Summary Table.

Selects the output events to be muxed for corresponding output, if that output event is active.

Figure 19-59. SSS_BLK1_2OUTSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLK2SIG2								BLK2SIG1								BLK1SIG2								BLK1SIG1							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 19-59. SSS_BLK1_2OUTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BLK2SIG2	R/W	0h	This word shows the association of the WADI output related BLK2-SIG2 to the related OUTEVTn. If the association bit is set then upon sequence word being active the corresponding bit of sequence word drives the related O/P. Where each bit is related to respective output event. Multiple bits can be set to seek the sequence as configured. 0x0 : No Safe state sequencer override. 0x1 : OUTEVT1 drives fixed value of corresponding bit 0x2 : OUTEVT2 drives fixed value of corresponding bit 0x4 : OUTEVT3drives fixed value of corresponding bit 0x8 : OUTEVT4drives fixed value of corresponding bit 0x10 : OUTEVT5 drives fixed value of corresponding bit 0x20 : OUTEVT6 drives fixed value of corresponding bit 0x40 : OUTEVT7 drives fixed value of corresponding bit 0x80 : OUTEVT8 drives fixed value of corresponding bit 0x3 : OUTEVT1,OUTEVT2 drive the sequence from respective bit 0x7 : OUTEVT1, OUTEVT2,OUTEVT3 drive the sequence from respective bit 0xF : OUTEVT1, OUTEVT2,OUTEVT3,OUTEVT4 drive the sequence from respective bit 0x6 : OUTEVT2,OUTEVT3 drive the sequence from respective bit 0xE : OUTEVT2,OUTEVT3,OUTEVT4 drive the sequence from respective bit 0xC : OUTEVT3,OUTEVT4 drive the sequence from respective bit Any such combinations of OUTEVTn are allowed depending upon sequence configuration settings. Only few depicted here. Reset type: SYSRSn

Table 19-59. SSS_BLK1_2OUTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-16	BLK2SIG1	R/W	0h	This word shows the association of the WADI output related BLK2-SIG1 to the related OUTEVTn. If the association bit is set then upon sequence word being active the corresponding bit of sequence word drives the related O/P. Where each bit is related to respective output event. Multiple bits can be set to seek the sequence as configured. 0x0 : No Safe state sequencer override. 0x1 : OUTEVT1 drives fixed value of corresponding bit 0x2 : OUTEVT2 drives fixed value of corresponding bit 0x4 : OUTEVT3drives fixed value of corresponding bit 0x8 : OUTEVT4drives fixed value of corresponding bit 0x10 : OUTEVT5 drives fixed value of corresponding bit 0x20 : OUTEVT6 drives fixed value of corresponding bit 0x40 : OUTEVT7 drives fixed value of corresponding bit 0x80 : OUTEVT8 drives fixed value of corresponding bit 0x3 : OUTEVT1,OUTEVT2 drive the sequence from respective bit 0x7 : OUTEVT1, OUTEVT2,OUTEVT3 drive the sequence from respective bit 0xF : OUTEVT1, OUTEVT2,OUTEVT3,OUTEVT4 drive the sequence from respective bit 0x6 : OUTEVT2,OUTEVT3 drive the sequence from respective bit 0xE : OUTEVT2,OUTEVT3,OUTEVT4 drive the sequence from respective bit 0xC : OUTEVT3,OUTEVT4 drive the sequence from respective bit Any such combinations of OUTEVTn are allowed depending upon sequence configuration settings. Only few depicted here. Reset type: SYSRSn
15-8	BLK1SIG2	R/W	0h	This word shows the association of the WADI output related BLK1-SIG2 to the related OUTEVTn. If the association bit is set then upon sequence word being active the corresponding bit of sequence word drives the related O/P. Where each bit is related to respective output event. Multiple bits can be set to seek the sequence as configured. 0x0 : No Safe state sequencer override. 0x1 : OUTEVT1 drives fixed value of corresponding bit 0x2 : OUTEVT2 drives fixed value of corresponding bit 0x4 : OUTEVT3drives fixed value of corresponding bit 0x8 : OUTEVT4drives fixed value of corresponding bit 0x10 : OUTEVT5 drives fixed value of corresponding bit 0x20 : OUTEVT6 drives fixed value of corresponding bit 0x40 : OUTEVT7 drives fixed value of corresponding bit 0x80 : OUTEVT8 drives fixed value of corresponding bit 0x3 : OUTEVT1,OUTEVT2 drive the sequence from respective bit 0x7 : OUTEVT1, OUTEVT2,OUTEVT3 drive the sequence from respective bit 0xF : OUTEVT1, OUTEVT2,OUTEVT3,OUTEVT4 drive the sequence from respective bit 0x6 : OUTEVT2,OUTEVT3 drive the sequence from respective bit 0xE : OUTEVT2,OUTEVT3,OUTEVT4 drive the sequence from respective bit 0xC : OUTEVT3,OUTEVT4 drive the sequence from respective bit Any such combinations of OUTEVTn are allowed depending upon sequence configuration settings. Only few depicted here. Reset type: SYSRSn

Table 19-59. SSS_BLK1_2OUTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	BLK1SIG1	R/W	0h	This word shows the association of the WADI output related BLK0-SIG1 to the related OUTEVTn. If the association bit is set then upon sequence word being active the corresponding bit of sequence word drives the related O/P. Where each bit is related to respective output event. Multiple bits can be set to seek the sequence as configured. 0x0 : No Safe state sequencer override. 0x1 : OUTEVT1 drives fixed value of corresponding bit 0x2 : OUTEVT2 drives fixed value of corresponding bit 0x4 : OUTEVT3drives fixed value of corresponding bit 0x8 : OUTEVT4drives fixed value of corresponding bit 0x10 : OUTEVT5 drives fixed value of corresponding bit 0x20 : OUTEVT6 drives fixed value of corresponding bit 0x40 : OUTEVT7 drives fixed value of corresponding bit 0x80 : OUTEVT8 drives fixed value of corresponding bit 0x3 : OUTEVT1,OUTEVT2 drive the sequence from respective bit 0x7 : OUTEVT1, OUTEVT2,OUTEVT3 drive the sequence from respective bit 0xF : OUTEVT1, OUTEVT2,OUTEVT3,OUTEVT4 drive the sequence from respective bit 0x6 : OUTEVT2,OUTEVT3 drive the sequence from respective bit 0xE : OUTEVT2,OUTEVT3,OUTEVT4 drive the sequence from respective bit 0xC : OUTEVT3,OUTEVT4 drive the sequence from respective bit Any such combinations of OUTEVTn are allowed depending upon sequence configuration settings. Only few depicted here. Reset type: SYSRSn

19.9.3.22 SSS_BLK3_4OUTSEL Register (Offset = C4h) [Reset = 00000000h]

SSS_BLK3_4OUTSEL is shown in Figure 19-60 and described in Table 19-60.

Return to the Summary Table.

Selects the output events to be muxed for corresponding output, if that output event is active.

Figure 19-60. SSS_BLK3_4OUTSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLK4SIG2								BLK4SIG1								BLK3SIG2								BLK3SIG1							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 19-60. SSS_BLK3_4OUTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BLK4SIG2	R/W	0h	This word shows the association of the WADI output related BLK4-SIG2 to the related OUTEVTn. If the association bit is set then upon sequence word being active the corresponding bit of sequence word drives the related O/P. Where each bit is related to respective output event. Multiple bits can be set to seek the sequence as configured. 0x0 : No Safe state sequencer override. 0x1 : OUTEVT1 drives fixed value of corresponding bit 0x2 : OUTEVT2 drives fixed value of corresponding bit 0x4 : OUTEVT3drives fixed value of corresponding bit 0x8 : OUTEVT4drives fixed value of corresponding bit 0x10 : OUTEVT5 drives fixed value of corresponding bit 0x20 : OUTEVT6 drives fixed value of corresponding bit 0x40 : OUTEVT7 drives fixed value of corresponding bit 0x80 : OUTEVT8 drives fixed value of corresponding bit 0x3 : OUTEVT1,OUTEVT2 drive the sequence from respective bit 0x7 : OUTEVT1, OUTEVT2,OUTEVT3 drive the sequence from respective bit 0xF : OUTEVT1, OUTEVT2,OUTEVT3,OUTEVT4 drive the sequence from respective bit 0x6 : OUTEVT2,OUTEVT3 drive the sequence from respective bit 0xE : OUTEVT2,OUTEVT3,OUTEVT4 drive the sequence from respective bit 0xC : OUTEVT3,OUTEVT4 drive the sequence from respective bit Any such combinations of OUTEVTn are allowed depending upon sequence configuration settings. Only few depicted here. Reset type: SYSRSn

Table 19-60. SSS_BLK3_4OUTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-16	BLK4SIG1	R/W	0h	This word shows the association of the WADI output related BLK4-SIG1 to the related OUTEVTn. If the association bit is set then upon sequence word being active the corresponding bit of sequence word drives the related O/P. Where each bit is related to respective output event. Multiple bits can be set to seek the sequence as configured. 0x0 : No Safe state sequencer override. 0x1 : OUTEVT1 drives fixed value of corresponding bit 0x2 : OUTEVT2 drives fixed value of corresponding bit 0x4 : OUTEVT3drives fixed value of corresponding bit 0x8 : OUTEVT4drives fixed value of corresponding bit 0x10 : OUTEVT5 drives fixed value of corresponding bit 0x20 : OUTEVT6 drives fixed value of corresponding bit 0x40 : OUTEVT7 drives fixed value of corresponding bit 0x80 : OUTEVT8 drives fixed value of corresponding bit 0x3 : OUTEVT1,OUTEVT2 drive the sequence from respective bit 0x7 : OUTEVT1, OUTEVT2,OUTEVT3 drive the sequence from respective bit 0xF : OUTEVT1, OUTEVT2,OUTEVT3,OUTEVT4 drive the sequence from respective bit 0x6 : OUTEVT2,OUTEVT3 drive the sequence from respective bit 0xE : OUTEVT2,OUTEVT3,OUTEVT4 drive the sequence from respective bit 0xC : OUTEVT3,OUTEVT4 drive the sequence from respective bit Any such combinations of OUTEVTn are allowed depending upon sequence configuration settings. Only few depicted here. Reset type: SYSRSn
15-8	BLK3SIG2	R/W	0h	This word shows the association of the WADI output related BLK3-SIG2 to the related OUTEVTn. If the association bit is set then upon sequence word being active the corresponding bit of sequence word drives the related O/P. Where each bit is related to respective output event. Multiple bits can be set to seek the sequence as configured. 0x0 : No Safe state sequencer override. 0x1 : OUTEVT1 drives fixed value of corresponding bit 0x2 : OUTEVT2 drives fixed value of corresponding bit 0x4 : OUTEVT3drives fixed value of corresponding bit 0x8 : OUTEVT4drives fixed value of corresponding bit 0x10 : OUTEVT5 drives fixed value of corresponding bit 0x20 : OUTEVT6 drives fixed value of corresponding bit 0x40 : OUTEVT7 drives fixed value of corresponding bit 0x80 : OUTEVT8 drives fixed value of corresponding bit 0x3 : OUTEVT1,OUTEVT2 drive the sequence from respective bit 0x7 : OUTEVT1, OUTEVT2,OUTEVT3 drive the sequence from respective bit 0xF : OUTEVT1, OUTEVT2,OUTEVT3,OUTEVT4 drive the sequence from respective bit 0x6 : OUTEVT2,OUTEVT3 drive the sequence from respective bit 0xE : OUTEVT2,OUTEVT3,OUTEVT4 drive the sequence from respective bit 0xC : OUTEVT3,OUTEVT4 drive the sequence from respective bit Any such combinations of OUTEVTn are allowed depending upon sequence configuration settings. Only few depicted here. Reset type: SYSRSn

Table 19-60. SSS_BLK3_4OUTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	BLK3SIG1	R/W	0h	This word shows the association of the WADI output related BLK3-SIG1 to the related OUTEVTn. If the association bit is set then upon sequence word being active the corresponding bit of sequence word drives the related O/P. Where each bit is related to respective output event. Multiple bits can be set to seek the sequence as configured. 0x0 : No Safe state sequencer override. 0x1 : OUTEVT1 drives fixed value of corresponding bit 0x2 : OUTEVT2 drives fixed value of corresponding bit 0x4 : OUTEVT3drives fixed value of corresponding bit 0x8 : OUTEVT4drives fixed value of corresponding bit 0x10 : OUTEVT5 drives fixed value of corresponding bit 0x20 : OUTEVT6 drives fixed value of corresponding bit 0x40 : OUTEVT7 drives fixed value of corresponding bit 0x80 : OUTEVT8 drives fixed value of corresponding bit 0x3 : OUTEVT1,OUTEVT2 drive the sequence from respective bit 0x7 : OUTEVT1, OUTEVT2,OUTEVT3 drive the sequence from respective bit 0xF : OUTEVT1, OUTEVT2,OUTEVT3,OUTEVT4 drive the sequence from respective bit 0x6 : OUTEVT2,OUTEVT3 drive the sequence from respective bit 0xE : OUTEVT2,OUTEVT3,OUTEVT4 drive the sequence from respective bit 0xC : OUTEVT3,OUTEVT4 drive the sequence from respective bit Any such combinations of OUTEVTn are allowed depending upon sequence configuration settings. Only few depicted here. Reset type: SYSRSn

19.9.3.23 SSS_OUTEVT1LINKCFG Register (Offset = E0h) [Reset = 00000000h]

SSS_OUTEVT1LINKCFG is shown in Figure 19-61 and described in Table 19-61.

Return to the Summary Table.

Output event1 configuration that Links it to other Output events.

Figure 19-61. SSS_OUTEVT1LINKCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OUTEVT1LINK							
R-0h								R/W-0h							

Table 19-61. SSS_OUTEVT1LINKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	OUTEVT1LINK	R/W	0h	This enables the configuration of linking EVTOUTn to EVTOUT1 0b0000_000X : No OUTEVT linking 0b0000_0010 : OUTEVT1 is linked to OUTEVT2 as its previous step in sequence this is used for 2 step cyclical sequence. 0b0000_0100 : OUTEVT1 is linked to OUTEVT3 as its previous step in sequence this is used for 3 step cyclical sequence. 0b0000_1000 : OUTEVT1 is linked to OUTEVT4 as its previous step in sequence this is used for 4 step cyclical sequence. 0b0001_0000 : OUTEVT1 is linked to OUTEVT5 as its previous step in sequence this is used for 5 step cyclical sequence. 0b0010_0000 : OUTEVT1 is linked to OUTEVT6 as its previous step in sequence this is used for 6 step cyclical sequence. 0b0100_0000 : OUTEVT1 is linked to OUTEVT7 as its previous step in sequence this is used for 7 step cyclical sequence. 0b1000_0000 : OUTEVT1 is linked to OUTEVT8 as its previous step in sequence this is used for 8 step cyclical sequence. Reset type: SYSRSn

19.9.3.24 SSS_OUTEVT2LINKCFG Register (Offset = E4h) [Reset = 00000000h]

SSS_OUTEVT2LINKCFG is shown in Figure 19-62 and described in Table 19-62.

Return to the Summary Table.

Output event2 configuration that Links it to other Output events.

Figure 19-62. SSS_OUTEVT2LINKCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OUTEVT2LINK							
R-0h								R/W-0h							

Table 19-62. SSS_OUTEVT2LINKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	OUTEVT2LINK	R/W	0h	This enables the configuration of linking EVTOUTn to EVTOUT2 0b0000_0000 : No OUTEVTn linking 0b0000_0001 : OUTEVT2 is linked to OUTEVT1 as its previous step in sequence this is used for multi step sequence. 0b0000_0100 : OUTEVT2 is linked to OUTEVT3 as its previous step in sequence this is used for 2 step cyclical sequence. Other : No OUTEVTn linking Reset type: SYSRSn

19.9.3.25 SSS_OUTEVT3LINKCFG Register (Offset = E8h) [Reset = 00000000h]

SSS_OUTEVT3LINKCFG is shown in Figure 19-63 and described in Table 19-63.

Return to the Summary Table.

Output event3 configuration that Links it to other Output events.

Figure 19-63. SSS_OUTEVT3LINKCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OUTEVT3LINK							
R-0h								R/W-0h							

Table 19-63. SSS_OUTEVT3LINKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	OUTEVT3LINK	R/W	0h	This enables the configuration of linking EVTOUTn to EVTOUT3 0b0000_0000 : No OUTEVTn linking 0b0000_0010 : OUTEVT3 is linked to OUTEVT2 as its previous step in sequence this is used for multi step sequence. 0b0000_1000 : OUTEVT3 is linked to OUTEVT4 as its previous step in sequence this is used for 2 step cyclical sequence. Other : No OUTEVTn dependency Reset type: SYSRSn

19.9.3.26 SSS_OUTEVT4LINKCFG Register (Offset = ECh) [Reset = 00000000h]

SSS_OUTEVT4LINKCFG is shown in Figure 19-64 and described in Table 19-64.

Return to the Summary Table.

Output event4 configuration that Links it to other Output events.

Figure 19-64. SSS_OUTEVT4LINKCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OUTEVT4LINK							
R-0h								R/W-0h							

Table 19-64. SSS_OUTEVT4LINKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	OUTEVT4LINK	R/W	0h	This enables the configuration of linking EVTOUTn to EVTOUT4 0b0000_0000 : No OUTEVTn linking 0b0000_0100 : OUTEVT4 is linked to OUTEVT3 as its previous step in sequence this is used for multi step sequence. Other : No OUTEVTn dependency Reset type: SYSRSn

19.9.3.27 SSS_OUTEVT5LINKCFG Register (Offset = F0h) [Reset = 00000000h]

SSS_OUTEVT5LINKCFG is shown in Figure 19-65 and described in Table 19-65.

Return to the Summary Table.

Output event5 configuration that Links it to other Output events.

Figure 19-65. SSS_OUTEVT5LINKCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OUTEVT5LINK							
R-0h								R/W-0h							

Table 19-65. SSS_OUTEVT5LINKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	OUTEVT5LINK	R/W	0h	This enables the configuration of linking EVTOUTn to EVTOUT5 0b0000_0000 : No OUTEVTn linking 0b0000_1000 : OUTEVT5 is linked to OUTEVT4 as its previous step in sequence this is used for multi step sequence. 0b0010_0000 : OUTEVT5 is linked to OUTEVT6 as its previous step in sequence this is used for 2 step cyclical sequence. 0b0100_0000 : OUTEVT5 is linked to OUTEVT6 as its previous step in sequence this is used for 3 step cyclical sequence. 0b1000_0000 : OUTEVT5 is linked to OUTEVT6 as its previous step in sequence this is used for 4 step cyclical sequence. Other : No OUTEVTn dependency Reset type: SYSRSn

19.9.3.28 SSS_OUTEVT6LINKCFG Register (Offset = F4h) [Reset = 00000000h]

SSS_OUTEVT6LINKCFG is shown in Figure 19-66 and described in Table 19-66.

Return to the Summary Table.

Output event6 configuration that Links it to other Output events.

Figure 19-66. SSS_OUTEVT6LINKCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OUTEVT6LINK							
R-0h								R/W-0h							

Table 19-66. SSS_OUTEVT6LINKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	OUTEVT6LINK	R/W	0h	This enables the configuration of linking EVTOUTn to EVTOUT6 0b0000_0000 : No OUTEVTn linking 0b0001_0000 : OUTEVT6 is linked to OUTEVT5 as its previous step in sequence this is used for multi step sequence. 0b0100_0000 : OUTEVT6 is linked to OUTEVT7 as its previous step in sequence this is used for 2 step cyclical sequence. Other : No OUTEVTn dependency Reset type: SYSRSn

19.9.3.29 SSS_OUTEVT7LINKCFG Register (Offset = F8h) [Reset = 00000000h]

SSS_OUTEVT7LINKCFG is shown in Figure 19-67 and described in Table 19-67.

Return to the Summary Table.

Output event7 configuration that Links it to other Output events.

Figure 19-67. SSS_OUTEVT7LINKCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OUTEVT7LINK							
R-0h								R/W-0h							

Table 19-67. SSS_OUTEVT7LINKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	OUTEVT7LINK	R/W	0h	This enables the configuration of linking EVTOUTn to EVTOUT7 0b0000_0000 : No OUTEVTn linking 0b0010_0000 : OUTEVT7 is linked to OUTEVT6 as its previous step in sequence this is used for multi step sequence. 0b1000_0000 : OUTEVT7 is linked to OUTEVT8 as its previous step in sequence this is used for 2 step cyclical sequence. Other : No OUTEVTn dependency Reset type: SYSRSn

19.9.3.30 SSS_OUTEVT8LINKCFG Register (Offset = FCh) [Reset = 00000000h]

SSS_OUTEVT8LINKCFG is shown in Figure 19-68 and described in Table 19-68.

Return to the Summary Table.

Output event8 configuration that Links it to other Output events.

Figure 19-68. SSS_OUTEVT8LINKCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								OUT8LINK							
R-0h																								R/W-0h							

Table 19-68. SSS_OUTEVT8LINKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	OUT8LINK	R/W	0h	This enables the configuration of linking EVTOUTn to EVTOUT8 0b0000_0000 : No OUTEVTn linking 0b0100_0000 : OUTEVT8 is linked to Sequence OUTEVT7 as its previous step in sequence this is used for multi step sequence. Other : No OUTEVTn dependency Reset type: SYSRSn

19.9.3.31 SSS_EVT1CFG Register (Offset = 100h) [Reset = 00000000h]

SSS_EVT1CFG is shown in Figure 19-69 and described in Table 19-69.

Return to the Summary Table.

Event1 used by SSS to trigger safe state sequence as per safe state sequencer configuration

Figure 19-69. SSS_EVT1CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EVT1							
R/W-0h																								R/W-0h							

Table 19-69. SSS_EVT1CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	Reserved
7-0	EVT1	R/W	0h	Event1 is compared with SSS_TRIGEVT is identification of match condition for set of bits configured '1' by user. Assertion of event is active high condition hence bits programmed '0' in this word have no effect of respective bits in SSS_TRIGEVT Once the match for the word is determined it either triggers SSS or advances to next trigger event as configured by SSS_TRIGEVT1_4CFG or SSS_TRIGEVT5_8CFG. [0] : SIG1 & SIGTOSIG event aggregation of block1 [1] : SIG2 event aggregation of block1 [2] : SIG1 & SIGTOSIG event aggregation of block2 [3] : SIG2 event aggregation of block2 [4] : SIG1 & SIGTOSIG event aggregation of block3 [5] : SIG2 event aggregation of block3 [6] : SIG1 & SIGTOSIG event aggregation of block4 [7] : SIG2 event aggregation of block4 Reset type: SYSRSn

19.9.3.32 SSS_EVT2CFG Register (Offset = 104h) [Reset = 00000000h]

SSS_EVT2CFG is shown in Figure 19-70 and described in Table 19-70.

Return to the Summary Table.

Event2 used by SSS to trigger safe state sequence as per safe state sequencer configuration

Figure 19-70. SSS_EVT2CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EVT2							
R/W-0h																								R/W-0h							

Table 19-70. SSS_EVT2CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	Reserved
7-0	EVT2	R/W	0h	Event2 is compared with SSS_TRIGEVT is identification of match condition for set of bits configured '1' by user. Assertion of event is active high condition hence bits programmed '0' in this word have no effect of respective bits in SSS_TRIGEVT Once the match for the word is determined it either triggers SSS or advances to next trigger event as configured by SSS_TRIGEVT1_4CFG or SSS_TRIGEVT5_8CFG. [0] : SIG1 & SIGTOSIG event aggregation of block1 [1] : SIG2 event aggregation of block1 [2] : SIG1 & SIGTOSIG event aggregation of block2 [3] : SIG2 event aggregation of block2 [4] : SIG1 & SIGTOSIG event aggregation of block3 [5] : SIG2 event aggregation of block3 [6] : SIG1 & SIGTOSIG event aggregation of block4 [7] : SIG2 event aggregation of block4 Reset type: SYSRSn

19.9.3.33 SSS_EVT3CFG Register (Offset = 108h) [Reset = 00000000h]

SSS_EVT3CFG is shown in Figure 19-71 and described in Table 19-71.

Return to the Summary Table.

Event3 used by SSS to trigger safe state sequence as per safe state sequencer configuration

Figure 19-71. SSS_EVT3CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVT3_BLK5_TO_16																								EVT3							
R/W-0h																								R/W-0h							

Table 19-71. SSS_EVT3CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	EVT3_BLK5_TO_16	R/W	0h	[8] : SIG1 & SIGTOSIG event aggregation of block5 [9] : SIG2 event aggregation of block5 [10] : SIG1 & SIGTOSIG event aggregation of block6 [11] : SIG2 event aggregation of block6 [12] : SIG1 & SIGTOSIG event aggregation of block7 [13] : SIG2 event aggregation of block7 [14] : SIG1 & SIGTOSIG event aggregation of block8 [15] : SIG2 event aggregation of block8 [16] : SIG1 & SIGTOSIG event aggregation of block9 [17] : SIG2 event aggregation of block9 [18] : SIG1 & SIGTOSIG event aggregation of block10 [19] : SIG2 event aggregation of block10 [20] : SIG1 & SIGTOSIG event aggregation of block11 [21] : SIG2 event aggregation of block11 [22] : SIG1 & SIGTOSIG event aggregation of block12 [23] : SIG2 event aggregation of block12 [24] : SIG1 & SIGTOSIG event aggregation of block13 [25] : SIG2 event aggregation of block13 [26] : SIG1 & SIGTOSIG event aggregation of block14 [27] : SIG2 event aggregation of block14 [28] : SIG1 & SIGTOSIG event aggregation of block15 [29] : SIG2 event aggregation of block15 [30] : SIG1 & SIGTOSIG event aggregation of block16 [31] : SIG2 event aggregation of block16 Reset type: SYSRSn
7-0	EVT3	R/W	0h	Event3 is compared with SSS_TRIGEVT is identification of match condition for set of bits configured '1' by user. Assertion of event is active high condition hence bits programmed '0' in this word have no effect of respective bits in SSS_TRIGEVT Once the match for the word is determined it either triggers SSS or advances to next trigger event as configured by SSS_TRIGEVT1_4CFG or SSS_TRIGEVT5_8CFG. [0] : SIG1 & SIGTOSIG event aggregation of block1 [1] : SIG2 event aggregation of block1 [2] : SIG1 & SIGTOSIG event aggregation of block2 [3] : SIG2 event aggregation of block2 [4] : SIG1 & SIGTOSIG event aggregation of block3 [5] : SIG2 event aggregation of block3 [6] : SIG1 & SIGTOSIG event aggregation of block4 [7] : SIG2 event aggregation of block4 Reset type: SYSRSn

19.9.3.34 SSS_EVT4CFG Register (Offset = 10Ch) [Reset = 00000000h]

SSS_EVT4CFG is shown in Figure 19-72 and described in Table 19-72.

Return to the Summary Table.

Event4 used by SSS to trigger safe state sequence as per safe state sequencer configuration

Figure 19-72. SSS_EVT4CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVT4_BLK5_TO_16																								EVT4							
R/W-0h																								R/W-0h							

Table 19-72. SSS_EVT4CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	EVT4_BLK5_TO_16	R/W	0h	[8] : SIG1 & SIGTOSIG event aggregation of block5 [9] : SIG2 event aggregation of block5 [10] : SIG1 & SIGTOSIG event aggregation of block6 [11] : SIG2 event aggregation of block6 [12] : SIG1 & SIGTOSIG event aggregation of block7 [13] : SIG2 event aggregation of block7 [14] : SIG1 & SIGTOSIG event aggregation of block8 [15] : SIG2 event aggregation of block8 [16] : SIG1 & SIGTOSIG event aggregation of block9 [17] : SIG2 event aggregation of block9 [18] : SIG1 & SIGTOSIG event aggregation of block10 [19] : SIG2 event aggregation of block10 [20] : SIG1 & SIGTOSIG event aggregation of block11 [21] : SIG2 event aggregation of block11 [22] : SIG1 & SIGTOSIG event aggregation of block12 [23] : SIG2 event aggregation of block12 [24] : SIG1 & SIGTOSIG event aggregation of block13 [25] : SIG2 event aggregation of block13 [26] : SIG1 & SIGTOSIG event aggregation of block14 [27] : SIG2 event aggregation of block14 [28] : SIG1 & SIGTOSIG event aggregation of block15 [29] : SIG2 event aggregation of block15 [30] : SIG1 & SIGTOSIG event aggregation of block16 [31] : SIG2 event aggregation of block16 Reset type: SYSRSn
7-0	EVT4	R/W	0h	Event4 is compared with SSS_TRIGEVT is identification of match condition for set of bits configured '1' by user. Assertion of event is active high condition hence bits programmed '0' in this word have no effect of respective bits in SSS_TRIGEVT Once the match for the word is determined it either triggers SSS or advances to next trigger event as configured by SSS_TRIGEVT1_4CFG or SSS_TRIGEVT5_8CFG. [0] : SIG1 & SIGTOSIG event aggregation of block1 [1] : SIG2 event aggregation of block1 [2] : SIG1 & SIGTOSIG event aggregation of block2 [3] : SIG2 event aggregation of block2 [4] : SIG1 & SIGTOSIG event aggregation of block3 [5] : SIG2 event aggregation of block3 [6] : SIG1 & SIGTOSIG event aggregation of block4 [7] : SIG2 event aggregation of block4 Reset type: SYSRSn

19.9.3.35 SSS_TRIGEVT1_4CFG Register (Offset = 110h) [Reset = 00000000h]

SSS_TRIGEVT1_4CFG is shown in Figure 19-73 and described in Table 19-73.

Return to the Summary Table.

Allows individual, sequenced and tandem use of event words to trigger the safe sequence

Figure 19-73. SSS_TRIGEVT1_4CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				TRIG_EVT4CFG				RESERVED				TRIG_EVT3CFG			
R-0h				R/W-0h				R-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				TRIG_EVT2CFG				RESERVED				TRIG_EVT1CFG			
R-0h				R/W-0h				R-0h				R/W-0h			

Table 19-73. SSS_TRIGEVT1_4CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Reserved
27-24	TRIG_EVT4CFG	R/W	0h	Based on configured value the Event 4 is used for trigger as independent trigger or in sequence with other words 0x0 : Event 4 is not used for any trigger 0x1 : Event 4 is independently used for trigger 0x9 : After Event 4 match sequence check moves to Event 5 is matched before next step. For 4 trigger event system this defaults to 0x1 i.e. triggers the sequencer if configured Any other value : Event 4 is not used for any trigger Reset type: SYSRSn
23-20	RESERVED	R	0h	Reserved
19-16	TRIG_EVT3CFG	R/W	0h	Based on configured value the Event 3 is used for trigger as independent trigger or in sequence with other words 0x0 : Event 3 is not used for any trigger 0x1 : Event 3 is independently used for trigger 0x9 : After Event 3 match sequence check moves to Event 4 is matched before next step Any other value : Event 3 is not used for any trigger Reset type: SYSRSn
15-12	RESERVED	R	0h	Reserved
11-8	TRIG_EVT2CFG	R/W	0h	Based on configured value the Event 2 is used for trigger as independent trigger or in sequence with other words 0x0 : Event 2 is not used for any trigger 0x1 : Event 2 is independently used for trigger 0x9 : After Event 2 match sequence check moves to Event 3 is matched before next step Any other value : Event 2 is not used for any trigger Reset type: SYSRSn
7-4	RESERVED	R	0h	Reserved
3-0	TRIG_EVT1CFG	R/W	0h	Based on configured value the Event 1 is used for trigger as independent trigger or in sequence with other words 0x0 : Event 1 is not used for any trigger 0x1 : Event 1 is independently used for trigger 0x9 : After Event 1 match sequence check moves to Event 2 is matched before next step Any other value : Event 1 is not used for any trigger Reset type: SYSRSn

19.9.3.36 SSS_BLKSOUTEVT1CFG Register (Offset = 180h) [Reset = 00000000h]

SSS_BLKSOUTEVT1CFG is shown in Figure 19-74 and described in Table 19-74.

Return to the Summary Table.

Output event1 used by SSS to drive WADI outputs as per Sequence configuration.

Figure 19-74. SSS_BLKSOUTEVT1CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								OUTEVT1							
R/W-0h																								R/W-0h							

Table 19-74. SSS_BLKSOUTEVT1CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	Reserved
7-0	OUTEVT1	R/W	0h	Output on event 1 (OUTEVT1) provides value to be applied on respective signal output when selected. Bits selectively drive the corresponding output line. [0] : SIG1 output override of block1 [1] : SIG2 output override of block1 [2] : SIG1 output override of block2 [3] : SIG2 output override of block2 [4] : SIG1 output override of block3 [5] : SIG2 output override of block3 [6] : SIG1 output override of block4 [7] : SIG2 output override of block4 Reset type: SYSRSn

19.9.3.37 SSS_BLKSOUTEVT2CFG Register (Offset = 184h) [Reset = 00000000h]

SSS_BLKSOUTEVT2CFG is shown in Figure 19-75 and described in Table 19-75.

Return to the Summary Table.

Output event2 used by SSS to drive WADI outputs as per Sequence configuration.

Figure 19-75. SSS_BLKSOUTEVT2CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								OUTEVT2							
R/W-0h																								R/W-0h							

Table 19-75. SSS_BLKSOUTEVT2CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	Reserved
7-0	OUTEVT2	R/W	0h	Output on event 2(OUTEVT2) provides value to be applied on respective signal output when selected. Bits selectively drive the corresponding output line. [0] : SIG1 output override of block1 [1] : SIG2 output override of block1 [2] : SIG1 output override of block2 [3] : SIG2 output override of block2 [4] : SIG1 output override of block3 [5] : SIG2 output override of block3 [6] : SIG1 output override of block4 [7] : SIG2 output override of block4 Reset type: SYSRSn

19.9.3.38 SSS_BLKSOUTEVT3CFG Register (Offset = 188h) [Reset = 00000000h]

SSS_BLKSOUTEVT3CFG is shown in Figure 19-76 and described in Table 19-76.

Return to the Summary Table.

Output event3 used by SSS to drive WADI outputs as per Sequence configuration.

Figure 19-76. SSS_BLKSOUTEVT3CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								OUTEVT3							
R/W-0h																								R/W-0h							

Table 19-76. SSS_BLKSOUTEVT3CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	Reserved
7-0	OUTEVT3	R/W	0h	Output on event 3(OUTEVT3) provides value to be applied on respective signal output when selected. Bits selectively drive the corresponding output line. [0] : SIG1 output override of block1 [1] : SIG2 output override of block1 [2] : SIG1 output override of block2 [3] : SIG2 output override of block2 [4] : SIG1 output override of block3 [5] : SIG2 output override of block3 [6] : SIG1 output override of block4 [7] : SIG2 output override of block4 Reset type: SYSRSn

19.9.3.39 SSS_BLKSOUTEVT4CFG Register (Offset = 18Ch) [Reset = 00000000h]

SSS_BLKSOUTEVT4CFG is shown in Figure 19-77 and described in Table 19-77.

Return to the Summary Table.

Output event4 used by SSS to drive WADI outputs as per Sequence configuration.

Figure 19-77. SSS_BLKSOUTEVT4CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								OUTEVT4							
R/W-0h																								R/W-0h							

Table 19-77. SSS_BLKSOUTEVT4CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	Reserved
7-0	OUTEVT4	R/W	0h	Output on event 4(OUTEVT4) provides value to be applied on respective signal output when selected. Bits selectively drive the corresponding output line. [0] : SIG1 output override of block1 [1] : SIG2 output override of block1 [2] : SIG1 output override of block2 [3] : SIG2 output override of block2 [4] : SIG1 output override of block3 [5] : SIG2 output override of block3 [6] : SIG1 output override of block4 [7] : SIG2 output override of block4 Reset type: SYSRSn

19.9.3.40 SSS_OUTEVT1TRIGCFG Register (Offset = 190h) [Reset = 00000000h]

SSS_OUTEVT1TRIGCFG is shown in Figure 19-78 and described in Table 19-78.

Return to the Summary Table.

Output event1 configuration that drives which trigger or link make output event active

Figure 19-78. SSS_OUTEVT1TRIGCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
EVT8TRIG	EVT7TRIG	EVT6TRIG	EVT5TRIG	EVT4TRIG	EVT3TRIG	EVT2TRIG	EVT1TRIG
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 19-78. SSS_OUTEVT1TRIGCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	EVT8TRIG	R/W	0h	When this control is set '1', then upon Event 8 Trigger assertion OUTEVT1 is set and the duration based on time defined at SSS_OUTEVT1DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 8 Trigger has no effect. Reset type: SYSRSn
6	EVT7TRIG	R/W	0h	When this control is set '1', then upon Event 7 Trigger assertion OUTEVT1 is set and the duration based on time defined at SSS_OUTEVT1DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 7 Trigger has no effect. Reset type: SYSRSn
5	EVT6TRIG	R/W	0h	When this control is set '1', then upon Event 6 Trigger assertion OUTEVT1 is set and the duration based on time defined at SSS_OUTEVT1DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 6 Trigger has no effect. Reset type: SYSRSn
4	EVT5TRIG	R/W	0h	When this control is set '1', then upon Event 5 Trigger assertion OUTEVT1 is set and the duration based on time defined at SSS_OUTEVT1DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 5 Trigger has no effect. Reset type: SYSRSn

Table 19-78. SSS_OUTEVT1TRIGCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	EVT4TRIG	R/W	0h	When this control is set '1', then upon Event 4 Trigger assertion OUTEVT1 is set and the duration based on time defined at SSS_OUTEVT1DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 4 Trigger has no effect. Reset type: SYSRSn
2	EVT3TRIG	R/W	0h	When this control is set '1', then upon Event 3 Trigger assertion OUTEVT1 is set and the duration based on time defined at SSS_OUTEVT1DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 3 Trigger has no effect. Reset type: SYSRSn
1	EVT2TRIG	R/W	0h	When this control is set '1', then upon Event 2 Trigger assertion OUTEVT1 is set and the duration based on time defined at SSS_OUTEVT1DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 2 Trigger has no effect. Reset type: SYSRSn
0	EVT1TRIG	R/W	0h	When this control is set '1', then upon Event 1 Trigger assertion OUTEVT1 is set and the duration based on time defined at SSS_OUTEVT1DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 1 Trigger has no effect. Reset type: SYSRSn

19.9.3.41 SSS_OUTEVT2TRIGCFG Register (Offset = 194h) [Reset = 00000000h]

SSS_OUTEVT2TRIGCFG is shown in Figure 19-79 and described in Table 19-79.

Return to the Summary Table.

Output event2 configuration that drives which trigger or link make sequence word active

Figure 19-79. SSS_OUTEVT2TRIGCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
EVT8TRIG	EVT7TRIG	EVT6TRIG	EVT5TRIG	EVT4TRIG	EVT3TRIG	EVT2TRIG	EVT1TRIG
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 19-79. SSS_OUTEVT2TRIGCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	EVT8TRIG	R/W	0h	When this control is set '1', then upon Event 8 Trigger assertion OUTEVT2 is set and the duration based on time defined at SSS_OUTEVT1DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 8 Trigger has no effect. Reset type: SYSRSn
6	EVT7TRIG	R/W	0h	When this control is set '1', then upon Event 7 Trigger assertion OUTEVT2 is set and the duration based on time defined at SSS_OUTEVT1DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 7 Trigger has no effect. Reset type: SYSRSn
5	EVT6TRIG	R/W	0h	When this control is set '1', then upon Event 6 Trigger assertion OUTEVT2 is set and the duration based on time defined at SSS_OUTEVT1DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 6 Trigger has no effect. Reset type: SYSRSn
4	EVT5TRIG	R/W	0h	When this control is set '1', then upon Event 5 Trigger assertion OUTEVT2 is set and the duration based on time defined at SSS_OUTEVT1DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 5 Trigger has no effect. Reset type: SYSRSn

Table 19-79. SSS_OUTEVT2TRIGCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	EVT4TRIG	R/W	0h	When this control is set '1', then upon Event 4 Trigger assertion OUTEVT2 is set and the duration based on time defined at SSS_OUTEVT1DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 4 Trigger has no effect. Reset type: SYSRSn
2	EVT3TRIG	R/W	0h	When this control is set '1', then upon Event 3 Trigger assertion OUTEVT2 is set and the duration based on time defined at SSS_OUTEVT1DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 3 Trigger has no effect. Reset type: SYSRSn
1	EVT2TRIG	R/W	0h	When this control is set '1', then upon Event 2 Trigger assertion OUTEVT2 is set and the duration based on time defined at SSS_OUTEVT1DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 2 Trigger has no effect. Reset type: SYSRSn
0	EVT1TRIG	R/W	0h	When this control is set '1', then upon Event 1 Trigger assertion OUTEVT2 is set and the duration based on time defined at SSS_OUTEVT1DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 1 Trigger has no effect. Reset type: SYSRSn

19.9.3.42 SSS_OUTEVT3TRIGCFG Register (Offset = 198h) [Reset = 00000000h]

SSS_OUTEVT3TRIGCFG is shown in Figure 19-80 and described in Table 19-80.

Return to the Summary Table.

Output event3 configuration that drives which trigger or link make output event active

Figure 19-80. SSS_OUTEVT3TRIGCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
EVT8TRIG	EVT7TRIG	EVT6TRIG	EVT5TRIG	EVT4TRIG	EVT3TRIG	EVT2TRIG	EVT1TRIG
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 19-80. SSS_OUTEVT3TRIGCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	EVT8TRIG	R/W	0h	When this control is set '1', then upon Event 8 Trigger assertion OUTEVT3 is set and the duration based on time defined at SSS_OUTEVT3DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 8 Trigger has no effect. Reset type: SYSRSn
6	EVT7TRIG	R/W	0h	When this control is set '1', then upon Event 7 Trigger assertion OUTEVT3 is set and the duration based on time defined at SSS_OUTEVT3DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 7 Trigger has no effect. Reset type: SYSRSn
5	EVT6TRIG	R/W	0h	When this control is set '1', then upon Event 6 Trigger assertion OUTEVT3 is set and the duration based on time defined at SSS_OUTEVT3DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 6 Trigger has no effect. Reset type: SYSRSn
4	EVT5TRIG	R/W	0h	When this control is set '1', then upon Event 5 Trigger assertion OUTEVT3 is set and the duration based on time defined at SSS_OUTEVT3DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 5 Trigger has no effect. Reset type: SYSRSn

Table 19-80. SSS_OUTEVT3TRIGCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	EVT4TRIG	R/W	0h	When this control is set '1', then upon Event 4 Trigger assertion OUTEVT3 is set and the duration based on time defined at SSS_OUTEVT3DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 4 Trigger has no effect. Reset type: SYSRSn
2	EVT3TRIG	R/W	0h	When this control is set '1', then upon Event 3 Trigger assertion OUTEVT3 is set and the duration based on time defined at SSS_OUTEVT3DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 3 Trigger has no effect. Reset type: SYSRSn
1	EVT2TRIG	R/W	0h	When this control is set '1', then upon Event 2 Trigger assertion OUTEVT3 is set and the duration based on time defined at SSS_OUTEVT3DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 2 Trigger has no effect. Reset type: SYSRSn
0	EVT1TRIG	R/W	0h	When this control is set '1', then upon Event 1 Trigger assertion OUTEVT3 is set and the duration based on time defined at SSS_OUTEVT3DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 1 Trigger has no effect. Reset type: SYSRSn

19.9.3.43 SSS_OUTEVT4TRIGCFG Register (Offset = 19Ch) [Reset = 00000000h]

SSS_OUTEVT4TRIGCFG is shown in Figure 19-81 and described in Table 19-81.

Return to the Summary Table.

Output event4 configuration that drives which trigger or link make output event active

Figure 19-81. SSS_OUTEVT4TRIGCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
EVT8TRIG	EVT7TRIG	EVT6TRIG	EVT5TRIG	EVT4TRIG	EVT3TRIG	EVT2TRIG	EVT1TRIG
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 19-81. SSS_OUTEVT4TRIGCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	EVT8TRIG	R/W	0h	When this control is set '1', then upon Event 8 Trigger assertion OUTEVT4 is set and the duration based on time defined at SSS_OUTEVT4DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 8 Trigger has no effect. Reset type: SYSRSn
6	EVT7TRIG	R/W	0h	When this control is set '1', then upon Event 7 Trigger assertion OUTEVT4 is set and the duration based on time defined at SSS_OUTEVT4DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 7 Trigger has no effect. Reset type: SYSRSn
5	EVT6TRIG	R/W	0h	When this control is set '1', then upon Event 6 Trigger assertion OUTEVT4 is set and the duration based on time defined at SSS_OUTEVT4DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 6 Trigger has no effect. Reset type: SYSRSn
4	EVT5TRIG	R/W	0h	When this control is set '1', then upon Event 5 Trigger assertion OUTEVT4 is set and the duration based on time defined at SSS_OUTEVT4DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 5 Trigger has no effect. Reset type: SYSRSn

Table 19-81. SSS_OUTEVT4TRIGCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	EVT4TRIG	R/W	0h	When this control is set '1', then upon Event 4 Trigger assertion OUTEVT4 is set and the duration based on time defined at SSS_OUTEVT4DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 4 Trigger has no effect. Reset type: SYSRSn
2	EVT3TRIG	R/W	0h	When this control is set '1', then upon Event 3 Trigger assertion OUTEVT4 is set and the duration based on time defined at SSS_OUTEVT4DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 3 Trigger has no effect. Reset type: SYSRSn
1	EVT2TRIG	R/W	0h	When this control is set '1', then upon Event 2 Trigger assertion OUTEVT4 is set and the duration based on time defined at SSS_OUTEVT4DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 2 Trigger has no effect. Reset type: SYSRSn
0	EVT1TRIG	R/W	0h	When this control is set '1', then upon Event 1 Trigger assertion OUTEVT4 is set and the duration based on time defined at SSS_OUTEVT4DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 1 Trigger has no effect. Reset type: SYSRSn

19.9.3.44 SSS_OUTEVT1DUR Register (Offset = 1A0h) [Reset = 00000000h]

SSS_OUTEVT1DUR is shown in Figure 19-82 and described in Table 19-82.

Return to the Summary Table.

Output event1 configuration that drives delay for applying output

Figure 19-82. SSS_OUTEVT1DUR Register

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

Table 19-82. SSS_OUTEVT1DUR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	TIME	R/W	0h	Delay count configured is count down to '0' for holding the OUTEVTn active. Upon reaching '0' the OUTEVTn is deactivated and count is reloaded with the configured value for next cyclical round or trigger. 0xFF_FFFF is special value when OUTEVTn is kept active continuously and counter does not count down. This allows defining terminal state of the sequence. Reset type: SYSRSn

19.9.3.45 SSS_OUTEVT2DUR Register (Offset = 1A4h) [Reset = 00000000h]

SSS_OUTEVT2DUR is shown in Figure 19-83 and described in Table 19-83.

Return to the Summary Table.

Output event2 configuration that drives delay for applying output

Figure 19-83. SSS_OUTEVT2DUR Register

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

Table 19-83. SSS_OUTEVT2DUR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	TIME	R/W	0h	Delay count configured is count down to '0' for holding the OUTEVTn active. Upon reaching '0' the OUTEVTn is deactivated and count is reloaded with the configured value for next cyclical round or trigger. 0xFF_FFFF is special value when OUTEVTn is kept active continuously and counter does not count down. This allows defining terminal state of the sequence. Reset type: SYSRSn

19.9.3.46 SSS_OUTEVT3DUR Register (Offset = 1A8h) [Reset = 00000000h]

SSS_OUTEVT3DUR is shown in Figure 19-84 and described in Table 19-84.

Return to the Summary Table.

Output event3 configuration that drives delay for applying output

Figure 19-84. SSS_OUTEVT3DUR Register

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

Table 19-84. SSS_OUTEVT3DUR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	TIME	R/W	0h	Delay count configured is count down to '0' for holding the OUTEVTn active. Upon reaching '0' the OUTEVTn is deactivated and count is reloaded with the configured value for next cyclical round or trigger. 0xFF_FFFF is special value when OUTEVTn is kept active continuously and counter does not count down. This allows defining terminal state of the sequence. Reset type: SYSRSn

19.9.3.47 SSS_OUTEVT4DUR Register (Offset = 1ACh) [Reset = 00000000h]

SSS_OUTEVT4DUR is shown in Figure 19-85 and described in Table 19-85.

Return to the Summary Table.

Output event4 configuration that drives delay for applying output

Figure 19-85. SSS_OUTEVT4DUR Register

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

Table 19-85. SSS_OUTEVT4DUR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	TIME	R/W	0h	Delay count configured is count down to '0' for holding the OUTEVTn active. Upon reaching '0' the OUTEVTn is deactivated and count is reloaded with the configured value for next cyclical round or trigger. 0xFF_FFFF is special value when OUTEVTn is kept active continuously and counter does not count down. This allows defining terminal state of the sequence. Reset type: SYSRSn

19.9.3.48 SSS_EVT5CFG Register (Offset = 200h) [Reset = 00000000h]

SSS_EVT5CFG is shown in Figure 19-86 and described in Table 19-86.

Return to the Summary Table.

Event5 used by SSS to trigger safe sequence as per safe state sequencer configuration

Figure 19-86. SSS_EVT5CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVT5_BLK5_TO_16																								EVT5							
R/W-0h																								R/W-0h							

Table 19-86. SSS_EVT5CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	EVT5_BLK5_TO_16	R/W	0h	[8] : SIG1 & SIGTOSIG event aggregation of block5 [9] : SIG2 event aggregation of block5 [10] : SIG1 & SIGTOSIG event aggregation of block6 [11] : SIG2 event aggregation of block6 [12] : SIG1 & SIGTOSIG event aggregation of block7 [13] : SIG2 event aggregation of block7 [14] : SIG1 & SIGTOSIG event aggregation of block8 [15] : SIG2 event aggregation of block8 [16] : SIG1 & SIGTOSIG event aggregation of block9 [17] : SIG2 event aggregation of block9 [18] : SIG1 & SIGTOSIG event aggregation of block10 [19] : SIG2 event aggregation of block10 [20] : SIG1 & SIGTOSIG event aggregation of block11 [21] : SIG2 event aggregation of block11 [22] : SIG1 & SIGTOSIG event aggregation of block12 [23] : SIG2 event aggregation of block12 [24] : SIG1 & SIGTOSIG event aggregation of block13 [25] : SIG2 event aggregation of block13 [26] : SIG1 & SIGTOSIG event aggregation of block14 [27] : SIG2 event aggregation of block14 [28] : SIG1 & SIGTOSIG event aggregation of block15 [29] : SIG2 event aggregation of block15 [30] : SIG1 & SIGTOSIG event aggregation of block16 [31] : SIG2 event aggregation of block16 Reset type: SYSRSn
7-0	EVT5	R/W	0h	Event5 is compared with SSS_TRIGEVT is identification of match condition for set of bits configured '1' by user. Assertion of event is active high condition hence bits programmed '0' in this word have no effect of respective bits in SSS_TRIGEVT Once the match for the word is determined it either triggers SSS or advances to next trigger event as configured by SSS_TRIGEVT1_4CFG or SSS_TRIGEVT5_8CFG. [0] : SIG1 & SIGTOSIG event aggregation of block1 [1] : SIG2 event aggregation of block1 [2] : SIG1 & SIGTOSIG event aggregation of block2 [3] : SIG2 event aggregation of block2 [4] : SIG1 & SIGTOSIG event aggregation of block3 [5] : SIG2 event aggregation of block3 [6] : SIG1 & SIGTOSIG event aggregation of block4 [7] : SIG2 event aggregation of block4 Reset type: SYSRSn

19.9.3.49 SSS_EVT6CFG Register (Offset = 204h) [Reset = 00000000h]

SSS_EVT6CFG is shown in Figure 19-87 and described in Table 19-87.

Return to the Summary Table.

Event6 used by SSS to trigger safe sequence as per safe state sequencer configuration

Figure 19-87. SSS_EVT6CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVT6_BLK5_TO_16																								EVT6							
R/W-0h																								R/W-0h							

Table 19-87. SSS_EVT6CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	EVT6_BLK5_TO_16	R/W	0h	[8] : SIG1 & SIGTOSIG event aggregation of block5 [9] : SIG2 event aggregation of block5 [10] : SIG1 & SIGTOSIG event aggregation of block6 [11] : SIG2 event aggregation of block6 [12] : SIG1 & SIGTOSIG event aggregation of block7 [13] : SIG2 event aggregation of block7 [14] : SIG1 & SIGTOSIG event aggregation of block8 [15] : SIG2 event aggregation of block8 [16] : SIG1 & SIGTOSIG event aggregation of block9 [17] : SIG2 event aggregation of block9 [18] : SIG1 & SIGTOSIG event aggregation of block10 [19] : SIG2 event aggregation of block10 [20] : SIG1 & SIGTOSIG event aggregation of block11 [21] : SIG2 event aggregation of block11 [22] : SIG1 & SIGTOSIG event aggregation of block12 [23] : SIG2 event aggregation of block12 [24] : SIG1 & SIGTOSIG event aggregation of block13 [25] : SIG2 event aggregation of block13 [26] : SIG1 & SIGTOSIG event aggregation of block14 [27] : SIG2 event aggregation of block14 [28] : SIG1 & SIGTOSIG event aggregation of block15 [29] : SIG2 event aggregation of block15 [30] : SIG1 & SIGTOSIG event aggregation of block16 [31] : SIG2 event aggregation of block16 Reset type: SYSRSn
7-0	EVT6	R/W	0h	Event6 is compared with SSS_TRIGEVT is identification of match condition for set of bits configured '1' by user. Assertion of event is active high condition hence bits programmed '0' in this word have no effect of respective bits in SSS_TRIGEVT Once the match for the word is determined it either triggers SSS or advances to next trigger event as configured by SSS_TRIGEVT1_4CFG or SSS_TRIGEVT5_8CFG. [0] : SIG1 & SIGTOSIG event aggregation of block1 [1] : SIG2 event aggregation of block1 [2] : SIG1 & SIGTOSIG event aggregation of block2 [3] : SIG2 event aggregation of block2 [4] : SIG1 & SIGTOSIG event aggregation of block3 [5] : SIG2 event aggregation of block3 [6] : SIG1 & SIGTOSIG event aggregation of block4 [7] : SIG2 event aggregation of block4 Reset type: SYSRSn

19.9.3.50 SSS_EVT7CFG Register (Offset = 208h) [Reset = 00000000h]

SSS_EVT7CFG is shown in Figure 19-88 and described in Table 19-88.

Return to the Summary Table.

Event7 used by SSS to trigger safe state sequence as per safe state sequencer configuration

Figure 19-88. SSS_EVT7CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVT7_BLK5_TO_16																								EVT7							
R/W-0h																								R/W-0h							

Table 19-88. SSS_EVT7CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	EVT7_BLK5_TO_16	R/W	0h	[8] : SIG1 & SIGTOSIG event aggregation of block5 [9] : SIG2 event aggregation of block5 [10] : SIG1 & SIGTOSIG event aggregation of block6 [11] : SIG2 event aggregation of block6 [12] : SIG1 & SIGTOSIG event aggregation of block7 [13] : SIG2 event aggregation of block7 [14] : SIG1 & SIGTOSIG event aggregation of block8 [15] : SIG2 event aggregation of block8 [16] : SIG1 & SIGTOSIG event aggregation of block9 [17] : SIG2 event aggregation of block9 [18] : SIG1 & SIGTOSIG event aggregation of block10 [19] : SIG2 event aggregation of block10 [20] : SIG1 & SIGTOSIG event aggregation of block11 [21] : SIG2 event aggregation of block11 [22] : SIG1 & SIGTOSIG event aggregation of block12 [23] : SIG2 event aggregation of block12 [24] : SIG1 & SIGTOSIG event aggregation of block13 [25] : SIG2 event aggregation of block13 [26] : SIG1 & SIGTOSIG event aggregation of block14 [27] : SIG2 event aggregation of block14 [28] : SIG1 & SIGTOSIG event aggregation of block15 [29] : SIG2 event aggregation of block15 [30] : SIG1 & SIGTOSIG event aggregation of block16 [31] : SIG2 event aggregation of block16 Reset type: SYSRSn
7-0	EVT7	R/W	0h	Event1 is compared with SSS_TRIGEVT is identification of match condition for set of bits configured '1' by user. Assertion of event is active high condition hence bits programmed '0' in this word have no effect of respective bits in SSS_TRIGEVT Once the match for the word is determined it either triggers SSS or advances to next trigger event as configured by SSS_TRIGEVT1_4CFG or SSS_TRIGEVT5_8CFG. [0] : SIG1 & SIGTOSIG event aggregation of block1 [1] : SIG2 event aggregation of block1 [2] : SIG1 & SIGTOSIG event aggregation of block2 [3] : SIG2 event aggregation of block2 [4] : SIG1 & SIGTOSIG event aggregation of block3 [5] : SIG2 event aggregation of block3 [6] : SIG1 & SIGTOSIG event aggregation of block4 [7] : SIG2 event aggregation of block4 Reset type: SYSRSn

19.9.3.51 SSS_EVT8CFG Register (Offset = 20Ch) [Reset = 00000000h]

SSS_EVT8CFG is shown in Figure 19-89 and described in Table 19-89.

Return to the Summary Table.

Event8 used by SSS to trigger safe state sequence as per safe state sequencer configuration

Figure 19-89. SSS_EVT8CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVT8_BLK5_TO_16																								EVT8							
R/W-0h																								R/W-0h							

Table 19-89. SSS_EVT8CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	EVT8_BLK5_TO_16	R/W	0h	[8] : SIG1 & SIGTOSIG event aggregation of block5 [9] : SIG2 event aggregation of block5 [10] : SIG1 & SIGTOSIG event aggregation of block6 [11] : SIG2 event aggregation of block6 [12] : SIG1 & SIGTOSIG event aggregation of block7 [13] : SIG2 event aggregation of block7 [14] : SIG1 & SIGTOSIG event aggregation of block8 [15] : SIG2 event aggregation of block8 [16] : SIG1 & SIGTOSIG event aggregation of block9 [17] : SIG2 event aggregation of block9 [18] : SIG1 & SIGTOSIG event aggregation of block10 [19] : SIG2 event aggregation of block10 [20] : SIG1 & SIGTOSIG event aggregation of block11 [21] : SIG2 event aggregation of block11 [22] : SIG1 & SIGTOSIG event aggregation of block12 [23] : SIG2 event aggregation of block12 [24] : SIG1 & SIGTOSIG event aggregation of block13 [25] : SIG2 event aggregation of block13 [26] : SIG1 & SIGTOSIG event aggregation of block14 [27] : SIG2 event aggregation of block14 [28] : SIG1 & SIGTOSIG event aggregation of block15 [29] : SIG2 event aggregation of block15 [30] : SIG1 & SIGTOSIG event aggregation of block16 [31] : SIG2 event aggregation of block16 Reset type: SYSRSn
7-0	EVT8	R/W	0h	Event1 is compared with SSS_TRIGEVT is identification of match condition for set of bits configured '1' by user. Assertion of event is active high condition hence bits programmed '0' in this word have no effect of respective bits in SSS_TRIGEVT Once the match for the word is determined it either triggers SSS or advances to next trigger event as configured by SSS_TRIGEVT1_4CFG or SSS_TRIGEVT5_8CFG. [0] : SIG1 & SIGTOSIG event aggregation of block1 [1] : SIG2 event aggregation of block1 [2] : SIG1 & SIGTOSIG event aggregation of block2 [3] : SIG2 event aggregation of block2 [4] : SIG1 & SIGTOSIG event aggregation of block3 [5] : SIG2 event aggregation of block3 [6] : SIG1 & SIGTOSIG event aggregation of block4 [7] : SIG2 event aggregation of block4 Reset type: SYSRSn

19.9.3.52 SSS_TRIGEVT5_8CFG Register (Offset = 210h) [Reset = 00000000h]

SSS_TRIGEVT5_8CFG is shown in Figure 19-90 and described in Table 19-90.

Return to the Summary Table.

Allows individual, sequenced and tandem use of event words to trigger the safe sequence

Figure 19-90. SSS_TRIGEVT5_8CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				TRIG_EVT8CFG				RESERVED				TRIG_EVT7CFG			
R-0h				R/W-0h				R-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				TRIG_EVT6CFG				RESERVED				TRIG_EVT5CFG			
R-0h				R/W-0h				R-0h				R/W-0h			

Table 19-90. SSS_TRIGEVT5_8CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Reserved
27-24	TRIG_EVT8CFG	R/W	0h	Based on configured value the Event 8 is used for trigger as independent trigger or in sequence with other words 0x0 : Event 8 is not used for any trigger 0x1 : Event 8 is independently used for trigger or it is followed up after Word-3 match to trigger SSS Any other value : Event 8 is not used for any trigger Reset type: SYSRSn
23-20	RESERVED	R	0h	Reserved
19-16	TRIG_EVT7CFG	R/W	0h	Based on configured value the Event 7 is used for trigger as independent trigger or in sequence with other words 0x0 : Event 7 is not used for any trigger 0x1 : Event 7 is independently used for trigger or it is followed up after Word-2 match to trigger SSS 0x9 : After Event 7 match sequence check moves to Event 8 is matched before next step Any other value : Event 7 is not used for any trigger Reset type: SYSRSn
15-12	RESERVED	R	0h	Reserved
11-8	TRIG_EVT6CFG	R/W	0h	Based on configured value the Event 6 is used for trigger as independent trigger or in sequence with other words. 0x0 : Event 6 is not used for any trigger. 0x1 : Event 6 is independently used for trigger or it is followed up after Word-1 match to trigger SSS 0x9 : After Event 6 match sequence check moves to Event 7 is matched before next step Any other value : Event 6 is not used for any trigger Reset type: SYSRSn
7-4	RESERVED	R	0h	Reserved
3-0	TRIG_EVT5CFG	R/W	0h	Based on configured value the Event 5 is used for trigger as independent trigger or in sequence with other words. 0x0 : Event 5 is not used for any trigger 0x1 : Event 5 is independently used for trigger 0x9 : After Event 5 match sequence check moves to Event 6 is matched before next step Any other value : Event 5 is not used for any trigger Reset type: SYSRSn

19.9.3.53 SSS_BLKSOUTEVT5CFG Register (Offset = 280h) [Reset = 00000000h]

SSS_BLKSOUTEVT5CFG is shown in Figure 19-91 and described in Table 19-91.

Return to the Summary Table.

Output event5 used by SSS to drive WADI outputs as per Sequence configuration.

Figure 19-91. SSS_BLKSOUTEVT5CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								OUTEVT5							
R/W-0h																								R/W-0h							

Table 19-91. SSS_BLKSOUTEVT5CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	Reserved
7-0	OUTEVT5	R/W	0h	Output on event 5(OUTEVT5) provides value to be applied on respective signal output when selected. Bits selectively drive the corresponding output line. [0] : SIG1 output override of block1 [1] : SIG2 output override of block1 [2] : SIG1 output override of block2 [3] : SIG2 output override of block2 [4] : SIG1 output override of block3 [5] : SIG2 output override of block3 [6] : SIG1 output override of block4 [7] : SIG2 output override of block4 Reset type: SYSRSn

19.9.3.54 SSS_BLKSOUTEVT6CFG Register (Offset = 284h) [Reset = 00000000h]

SSS_BLKSOUTEVT6CFG is shown in Figure 19-92 and described in Table 19-92.

Return to the Summary Table.

Output event6 used by SSS to drive WADI outputs as per Sequence configuration.

Figure 19-92. SSS_BLKSOUTEVT6CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								OUTEVT6							
R/W-0h																								R/W-0h							

Table 19-92. SSS_BLKSOUTEVT6CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	Reserved
7-0	OUTEVT6	R/W	0h	Output on event 6(OUTEVT6) provides value to be applied on respective signal output when selected. Bits selectively drive the corresponding output line. [0] : SIG1 output override of block1 [1] : SIG2 output override of block1 [2] : SIG1 output override of block2 [3] : SIG2 output override of block2 [4] : SIG1 output override of block3 [5] : SIG2 output override of block3 [6] : SIG1 output override of block4 [7] : SIG2 output override of block4 Reset type: SYSRSn

19.9.3.55 SSS_BLKSOUTEVT7CFG Register (Offset = 288h) [Reset = 00000000h]

SSS_BLKSOUTEVT7CFG is shown in Figure 19-93 and described in Table 19-93.

Return to the Summary Table.

Output event7 used by SSS to drive WADI outputs as per Sequence configuration.

Figure 19-93. SSS_BLKSOUTEVT7CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								OUTEVT7							
R/W-0h																								R/W-0h							

Table 19-93. SSS_BLKSOUTEVT7CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	Reserved
7-0	OUTEVT7	R/W	0h	Output on event 7(OUTEVT7) provides value to be applied on respective signal output when selected. Bits selectively drive the corresponding output line. [0] : SIG1 output override of block1 [1] : SIG2 output override of block1 [2] : SIG1 output override of block2 [3] : SIG2 output override of block2 [4] : SIG1 output override of block3 [5] : SIG2 output override of block3 [6] : SIG1 output override of block4 [7] : SIG2 output override of block4 Reset type: SYSRSn

19.9.3.56 SSS_BLKSOUTEVT8CFG Register (Offset = 28Ch) [Reset = 00000000h]

SSS_BLKSOUTEVT8CFG is shown in Figure 19-94 and described in Table 19-94.

Return to the Summary Table.

Output event8 used by SSS to drive WADI outputs as per Sequence configuration.

Figure 19-94. SSS_BLKSOUTEVT8CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								OUTEVT8							
R/W-0h																								R/W-0h							

Table 19-94. SSS_BLKSOUTEVT8CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	Reserved
7-0	OUTEVT8	R/W	0h	Output on event 8(OUTEVT8) provides value to be applied on respective signal output when selected. Bits selectively drive the corresponding output line. [0] : SIG1 output override of block1 [1] : SIG2 output override of block1 [2] : SIG1 output override of block2 [3] : SIG2 output override of block2 [4] : SIG1 output override of block3 [5] : SIG2 output override of block3 [6] : SIG1 output override of block4 [7] : SIG2 output override of block4 Reset type: SYSRSn

19.9.3.57 SSS_OUTEVT5TRIGCFG Register (Offset = 290h) [Reset = 00000000h]

SSS_OUTEVT5TRIGCFG is shown in Figure 19-95 and described in Table 19-95.

Return to the Summary Table.

Output event5 configuration that drives which trigger or link make sequence word active

Figure 19-95. SSS_OUTEVT5TRIGCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
EVT8TRIG	EVT7TRIG	EVT6TRIG	EVT5TRIG	EVT4TRIG	EVT3TRIG	EVT2TRIG	EVT1TRIG
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 19-95. SSS_OUTEVT5TRIGCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	EVT8TRIG	R/W	0h	When this control is set '1', then upon Event 8 Trigger assertion OUTEVT5 is set and the duration based on time defined at SSS_OUTEVT5DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 8 Trigger has no effect. Reset type: SYSRSn
6	EVT7TRIG	R/W	0h	When this control is set '1', then upon Event 7 Trigger assertion OUTEVT5 is set and the duration based on time defined at SSS_OUTEVT5DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 7 Trigger has no effect. Reset type: SYSRSn
5	EVT6TRIG	R/W	0h	When this control is set '1', then upon Event 6 Trigger assertion OUTEVT5 is set and the duration based on time defined at SSS_OUTEVT5DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 6 Trigger has no effect. Reset type: SYSRSn
4	EVT5TRIG	R/W	0h	When this control is set '1', then upon Event 5 Trigger assertion OUTEVT5 is set and the duration based on time defined at SSS_OUTEVT5DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 5 Trigger has no effect. Reset type: SYSRSn

Table 19-95. SSS_OUTEVT5TRIGCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	EVT4TRIG	R/W	0h	When this control is set '1', then upon Event 4 Trigger assertion OUTEVT5 is set and the duration based on time defined at SSS_OUTEVT5DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 4 Trigger has no effect. Reset type: SYSRSn
2	EVT3TRIG	R/W	0h	When this control is set '1', then upon Event 3 Trigger assertion OUTEVT5 is set and the duration based on time defined at SSS_OUTEVT5DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 3 Trigger has no effect. Reset type: SYSRSn
1	EVT2TRIG	R/W	0h	When this control is set '1', then upon Event 2 Trigger assertion OUTEVT5 is set and the duration based on time defined at SSS_OUTEVT5DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 2 Trigger has no effect. Reset type: SYSRSn
0	EVT1TRIG	R/W	0h	When this control is set '1', then upon Event 1 Trigger assertion OUTEVT5 is set and the duration based on time defined at SSS_OUTEVT5DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 1 Trigger has no effect. Reset type: SYSRSn

19.9.3.58 SSS_OUTEVT6TRIGCFG Register (Offset = 294h) [Reset = 00000000h]

SSS_OUTEVT6TRIGCFG is shown in Figure 19-96 and described in Table 19-96.

Return to the Summary Table.

Output event6 configuration that drives which trigger or link make sequence word active

Figure 19-96. SSS_OUTEVT6TRIGCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
EVT8TRIG	EVT7TRIG	EVT6TRIG	EVT5TRIG	EVT4TRIG	EVT3TRIG	EVT2TRIG	EVT1TRIG
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 19-96. SSS_OUTEVT6TRIGCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	EVT8TRIG	R/W	0h	When this control is set '1', then upon Event 8 Trigger assertion OUTEVT6 is set and the duration based on time defined at SSS_OUTEVT6DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 8 Trigger has no effect. Reset type: SYSRSn
6	EVT7TRIG	R/W	0h	When this control is set '1', then upon Event 7 Trigger assertion OUTEVT6 is set and the duration based on time defined at SSS_OUTEVT6DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 7 Trigger has no effect. Reset type: SYSRSn
5	EVT6TRIG	R/W	0h	When this control is set '1', then upon Event 6 Trigger assertion OUTEVT6 is set and the duration based on time defined at SSS_OUTEVT6DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 6 Trigger has no effect. Reset type: SYSRSn
4	EVT5TRIG	R/W	0h	When this control is set '1', then upon Event 5 Trigger assertion OUTEVT6 is set and the duration based on time defined at SSS_OUTEVT6DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 5 Trigger has no effect. Reset type: SYSRSn

Table 19-96. SSS_OUTEVT6TRIGCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	EVT4TRIG	R/W	0h	When this control is set '1', then upon Event 4 Trigger assertion OUTEVT6 is set and the duration based on time defined at SSS_OUTEVT6DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 4 Trigger has no effect. Reset type: SYSRSn
2	EVT3TRIG	R/W	0h	When this control is set '1', then upon Event 3 Trigger assertion OUTEVT6 is set and the duration based on time defined at SSS_OUTEVT6DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 3 Trigger has no effect. Reset type: SYSRSn
1	EVT2TRIG	R/W	0h	When this control is set '1', then upon Event 2 Trigger assertion OUTEVT6 is set and the duration based on time defined at SSS_OUTEVT6DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 2 Trigger has no effect. Reset type: SYSRSn
0	EVT1TRIG	R/W	0h	When this control is set '1', then upon Event 1 Trigger assertion OUTEVT6 is set and the duration based on time defined at SSS_OUTEVT6DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 1 Trigger has no effect. Reset type: SYSRSn

19.9.3.59 SSS_OUTEVT7TRIGCFG Register (Offset = 298h) [Reset = 00000000h]

SSS_OUTEVT7TRIGCFG is shown in Figure 19-97 and described in Table 19-97.

Return to the Summary Table.

Output event7 configuration that drives which trigger or link make sequence word active

Figure 19-97. SSS_OUTEVT7TRIGCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
EVT8TRIG	EVT7TRIG	EVT6TRIG	EVT5TRIG	EVT4TRIG	EVT3TRIG	EVT2TRIG	EVT1TRIG
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 19-97. SSS_OUTEVT7TRIGCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	EVT8TRIG	R/W	0h	When this control is set '1', then upon Event 8 Trigger assertion OUTEVT7 is set and the duration based on time defined at SSS_OUTEVT7DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 8 Trigger has no effect. Reset type: SYSRSn
6	EVT7TRIG	R/W	0h	When this control is set '1', then upon Event 7 Trigger assertion OUTEVT7 is set and the duration based on time defined at SSS_OUTEVT7DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 7 Trigger has no effect. Reset type: SYSRSn
5	EVT6TRIG	R/W	0h	When this control is set '1', then upon Event 6 Trigger assertion OUTEVT7 is set and the duration based on time defined at SSS_OUTEVT7DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 6 Trigger has no effect. Reset type: SYSRSn
4	EVT5TRIG	R/W	0h	When this control is set '1', then upon Event 5 Trigger assertion OUTEVT7 is set and the duration based on time defined at SSS_OUTEVT7DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 5 Trigger has no effect. Reset type: SYSRSn

Table 19-97. SSS_OUTEVT7TRIGCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	EVT4TRIG	R/W	0h	When this control is set '1', then upon Event 4 Trigger assertion OUTEVT7 is set and the duration based on time defined at SSS_OUTEVT7DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 4 Trigger has no effect. Reset type: SYSRSn
2	EVT3TRIG	R/W	0h	When this control is set '1', then upon Event 3 Trigger assertion OUTEVT7 is set and the duration based on time defined at SSS_OUTEVT7DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 3 Trigger has no effect. Reset type: SYSRSn
1	EVT2TRIG	R/W	0h	When this control is set '1', then upon Event 2 Trigger assertion OUTEVT7 is set and the duration based on time defined at SSS_OUTEVT7DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 2 Trigger has no effect. Reset type: SYSRSn
0	EVT1TRIG	R/W	0h	When this control is set '1', then upon Event 1 Trigger assertion OUTEVT7 is set and the duration based on time defined at SSS_OUTEVT7DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 1 Trigger has no effect. Reset type: SYSRSn

19.9.3.60 SSS_OUTEVT8TRIGCFG Register (Offset = 29Ch) [Reset = 00000000h]

SSS_OUTEVT8TRIGCFG is shown in Figure 19-98 and described in Table 19-98.

Return to the Summary Table.

Output event8 configuration that drives which trigger or link make sequence word active

Figure 19-98. SSS_OUTEVT8TRIGCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
EVT8TRIG	EVT7TRIG	EVT6TRIG	EVT5TRIG	EVT4TRIG	EVT3TRIG	EVT2TRIG	EVT1TRIG
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 19-98. SSS_OUTEVT8TRIGCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	EVT8TRIG	R/W	0h	When this control is set '1', then upon Event 8 Trigger assertion OUTEVT8 is set and the duration based on time defined at SSS_OUTEVT8DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 8 Trigger has no effect. Reset type: SYSRSn
6	EVT7TRIG	R/W	0h	When this control is set '1', then upon Event 7 Trigger assertion OUTEVT8 is set and the duration based on time defined at SSS_OUTEVT8DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 7 Trigger has no effect. Reset type: SYSRSn
5	EVT6TRIG	R/W	0h	When this control is set '1', then upon Event 6 Trigger assertion OUTEVT8 is set and the duration based on time defined at SSS_OUTEVT8DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 6 Trigger has no effect. Reset type: SYSRSn
4	EVT5TRIG	R/W	0h	When this control is set '1', then upon Event 5 Trigger assertion OUTEVT8 is set and the duration based on time defined at SSS_OUTEVT8DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 5 Trigger has no effect. Reset type: SYSRSn

Table 19-98. SSS_OUTEVT8TRIGCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	EVT4TRIG	R/W	0h	When this control is set '1', then upon Event 4 Trigger assertion OUTEVT8 is set and the duration based on time defined at SSS_OUTEVT8DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 4 Trigger has no effect. Reset type: SYSRSn
2	EVT3TRIG	R/W	0h	When this control is set '1', then upon Event 3 Trigger assertion OUTEVT8 is set and the duration based on time defined at SSS_OUTEVT8DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 3 Trigger has no effect. Reset type: SYSRSn
1	EVT2TRIG	R/W	0h	When this control is set '1', then upon Event 2 Trigger assertion OUTEVT8 is set and the duration based on time defined at SSS_OUTEVT8DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 2 Trigger has no effect. Reset type: SYSRSn
0	EVT1TRIG	R/W	0h	When this control is set '1', then upon Event 1 Trigger assertion OUTEVT8 is set and the duration based on time defined at SSS_OUTEVT8DUR[TIME] If the trigger is asserted (rise edge) while counter is active then counter is restarted. Hence overlapping triggers extends the duration for which OUTEVTn is held. If control set to '0' Event 1 Trigger has no effect. Reset type: SYSRSn

19.9.3.61 SSS_OUTEVT5DUR Register (Offset = 2A0h) [Reset = 00000000h]

SSS_OUTEVT5DUR is shown in Figure 19-99 and described in Table 19-99.

Return to the Summary Table.

Output event5 configuration that drives delay for applying output

Figure 19-99. SSS_OUTEVT5DUR Register

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

Table 19-99. SSS_OUTEVT5DUR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	TIME	R/W	0h	Delay count configured is count down to '0' for holding the OUTEVTn active. Upon reaching '0' the OUTEVTn is deactivated and count is reloaded with the configured value for next cyclical round or trigger. 0xFF_FFFF is special value when OUTEVTn is kept active continuously and counter does not count down. This allows defining terminal state of the sequence. Reset type: SYSRSn

19.9.3.62 SSS_OUTEVT6DUR Register (Offset = 2A4h) [Reset = 00000000h]

SSS_OUTEVT6DUR is shown in Figure 19-100 and described in Table 19-100.

Return to the Summary Table.

Output event6 configuration that drives delay for applying output

Figure 19-100. SSS_OUTEVT6DUR Register

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

Table 19-100. SSS_OUTEVT6DUR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	TIME	R/W	0h	Delay count configured is count down to '0' for holding the OUTEVTn active. Upon reaching '0' the OUTEVTn is deactivated and count is reloaded with the configured value for next cyclical round or trigger. 0xFF_FFFF is special value when OUTEVTn is kept active continuously and counter does not count down. This allows defining terminal state of the sequence. Reset type: SYSRSn

19.9.3.63 SSS_OUTEVT7DUR Register (Offset = 2A8h) [Reset = 00000000h]

SSS_OUTEVT7DUR is shown in Figure 19-101 and described in Table 19-101.

Return to the Summary Table.

Output event7 configuration that drives delay for applying output

Figure 19-101. SSS_OUTEVT7DUR Register

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

Table 19-101. SSS_OUTEVT7DUR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	TIME	R/W	0h	Delay count configured is count down to '0' for holding the OUTEVTn active. Upon reaching '0' the OUTEVTn is deactivated and count is reloaded with the configured value for next cyclical round or trigger. 0xFF_FFFF is special value when OUTEVTn is kept active continuously and counter does not count down. This allows defining terminal state of the sequence. Reset type: SYSRSn

19.9.3.64 SSS_OUTEVT8DUR Register (Offset = 2ACh) [Reset = 00000000h]

SSS_OUTEVT8DUR is shown in Figure 19-102 and described in Table 19-102.

Return to the Summary Table.

Output event8 configuration that drives delay for applying output

Figure 19-102. SSS_OUTEVT8DUR Register

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

Table 19-102. SSS_OUTEVT8DUR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	TIME	R/W	0h	Delay count configured is count down to '0' for holding the OUTEVTn active. Upon reaching '0' the OUTEVTn is deactivated and count is reloaded with the configured value for next cyclical round or trigger. 0xFF_FFFF is special value when OUTEVTn is kept active continuously and counter does not count down. This allows defining terminal state of the sequence. Reset type: SYSRSn

19.9.3.65 PARTEST Register (Offset = 2C8h) [Reset = 00000000h]

PARTEST is shown in Figure 19-103 and described in Table 19-103.

Return to the Summary Table.

Enables parity test

Figure 19-103. PARTEST Register

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

Table 19-103. PARTEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	TESTEN	R/W	0h	1010: Parity test feature is enabled All other values: Parity test feature is disabled Note: (1) When the parity test feature is enabled, actual registers are not accessible in the memory map. Instead, the parity values (final XOR output indicating the parity error) are accessible Parity is computed for every byte and the corresponding parity error value is available at the bit-0 of every byte. Value of '1' written to the parity bit after enabling the parity test feature can be used to inject the error by inverting the stored parity value (2) It is recommended to leave the field as 0101 or 0000 after completing the parity test Reset type: SYSRSn

Chapter 20

Crossbar (X-BAR)

The crossbars (referred to as X-BAR throughout this chapter) provide flexibility to connect device inputs, outputs, and internal resources in a variety of configurations.

The device contains a total of six X-BARs:

- Input X-BAR
- Output X-BAR
- CLB X-BAR
- ePWM X-BAR
- MINDB X-BAR
- ICL X-BAR

Each of the X-BARs is named according to where the X-BAR takes signals. For example, the Input X-BAR brings external signals “in” to the device. The Output X-BAR takes internal signals “out” of the device to a GPIO. The ePWM X-BAR takes signals to the ePWM modules.

20.1 X-BAR Related Collateral

Foundational Materials

- C29x Academy - Crossbars (X-BAR)

20.2 Input X-BAR, ICL XBAR, MINDB XBAR,

On this device, the Input X-BAR is used to route signals from a GPIO to many different IP blocks such as the ADC, eCAP, ePWM, and external interrupts. The input of each Input X-BAR instance (INPUTx) can be any GPIO, while the output of each instance connects to various IP blocks in the device. The digital input of AIOs are also available as inputs to the Input X-BAR. This flexibility relieves some of the constraints on peripheral muxing by allowing the user to connect any GPIO to the specified outputs of each Input X-BAR instance. Note that the GPIO selected by the Input X-BAR can be configured as either an input or an output. The Input X-BAR simply connects the signal on the input buffer to the output of the selected Input X-BAR instance. Therefore, you can do things such as route the output of an ePWM to the eCAP module for a frequency test).

The Input X-BAR is configured by way of the INPUTxSELECT registers. The destinations for each INPUTx are shown in Figure 20-1 and Table 20-1. For additional details on how each Input X-BAR connects to other IP blocks throughout the device, look for references to Input X-BAR in the chapter associated with that IP. Note that the destinations of each INPUTx are fixed and are not user-configurable. For more information on configuring the Input X-BAR, see the INPUT_XBAR_REGS register definitions in the *XBAR Registers* section.

Note

All input sources to the generic XBAR can be active high.

The minimum input pulse width required for ePWM, CLB XBAR (CLB Clocks required) and SYSCLK for Output XBAR is 3 ticks of the respective clocks.

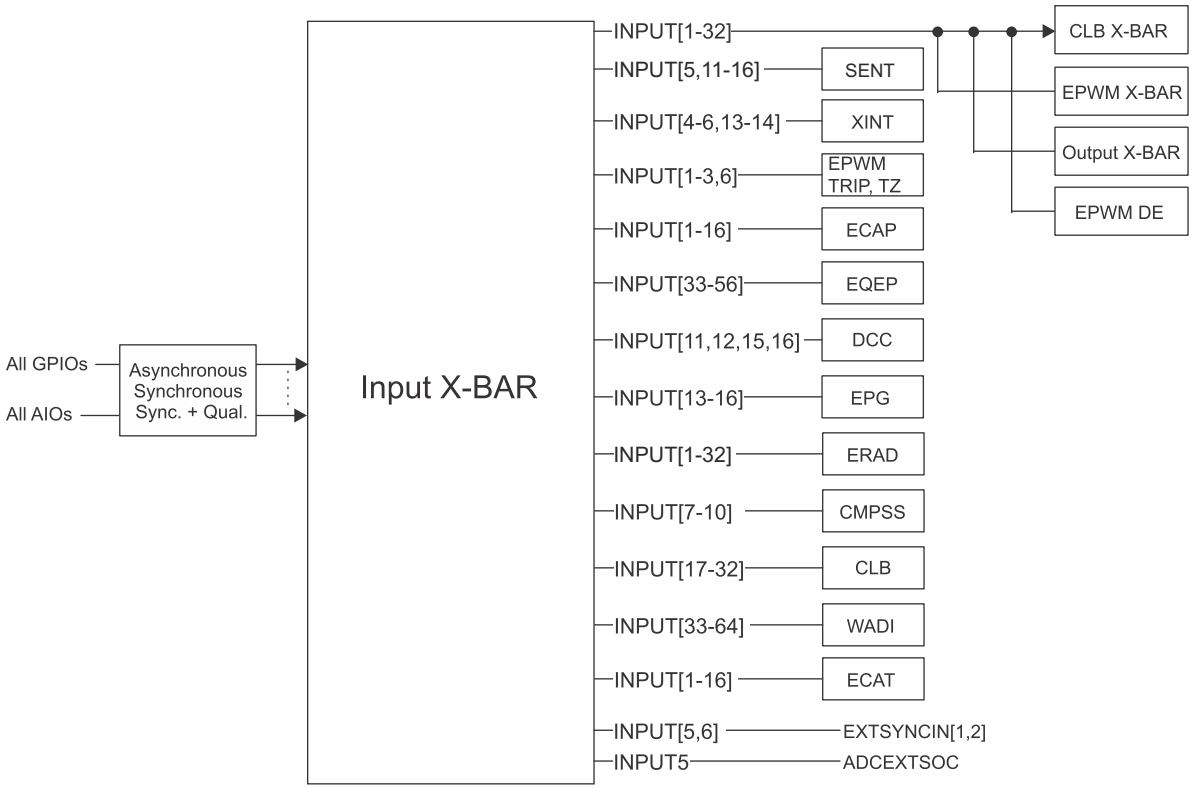


Figure 20-1. Input X-BAR

Note

INPUTXBARx, INPUTXBAR_INPUTx, and INPUTx (when referenced in the context of Input X-BAR) are equivalent in all C2000 software and documentation.

Table 20-1. Input X-BAR Destinations

INPUT	EPWMXBAR	CLBXBAR	OUTPUTXBAR	XINT	EPWM TRIP	EPWMDEL	ECAP	EQEP	ADCSOC	SYNCIN	DCCx	EPG	ERAD	CMPSS	CLB	WADI1	WADI2	ECAT	SENT
1	Yes	Yes	Yes	-	TZ1,TRIP1	Yes	Yes	-	-	-	-	-	Yes	-	-	-	-	Yes	-
2	Yes	Yes	Yes	-	TZ2,TRIP2	Yes	Yes	-	-	-	-	-	Yes	-	-	-	-	Yes	-
3	Yes	Yes	Yes	-	TZ3,TRIP3	Yes	Yes	-	-	-	-	-	Yes	-	-	-	-	Yes	-
4	Yes	Yes	Yes	XINT1	-	Yes	Yes	-	-	-	-	-	Yes	-	-	-	-	Yes	-
5	Yes	Yes	Yes	XINT2	-	Yes	Yes	-	ADCEXTSOC	EXTSYNCIN1	-	-	Yes	-	-	-	-	Yes	Yes
6	Yes	Yes	Yes	XINT3	TRIP6	Yes	Yes	-	-	EXTSYNCIN2	-	-	Yes	-	-	-	-	Yes	-
7	Yes	Yes	Yes	-	-	Yes	Yes	-	-	-	-	-	Yes	CMPSS1_EXT_FILTIN_H,CMPSS3_EXT_FILTIN_H	-	-	-	Yes	-
8	Yes	Yes	Yes	-	-	Yes	Yes	-	-	-	-	-	Yes	CMPSS1_EXT_FILTIN_L ,CMPSS3_EXT_FILTIN_L	-	-	-	Yes	-
9	Yes	Yes	Yes	-	-	Yes	Yes	-	-	-	-	-	Yes	CMPSS2_EXT_FILTIN_H,CMPSS4_EXT_FILTIN_H	-	-	-	Yes	-
10	Yes	Yes	Yes	-	-	Yes	Yes	-	-	-	-	-	Yes	CMPSS2_EXT_FILTIN_L ,CMPSS4_EXT_FILTIN_L	-	-	-	Yes	-
11	Yes	Yes	Yes	-	-	Yes	Yes	-	-	-	CLK1	-	Yes	-	-	-	-	Yes	Yes

Table 20-1. Input X-BAR Destinations (continued)

INPUT	EPWMXBAR	CLBXBAR	OUTPUTXBAR	XINT	EPWM TRIP	EPWMDEL	ECAP	EQEP	ADCSOC	SYNCIN	DCCx	EPG	ERAD	CMPSS	CLB	WADI1	WADI2	ECAT	SENT
12	Yes	Yes	Yes	-	-	Yes	Yes	-	-	-	CLK1	-	Yes	-	-	-	-	Yes	Yes
13	Yes	Yes	Yes	XINT4	-	Yes	Yes	-	-	-	-	EPG1IN1	Yes	-	-	-	-	Yes	Yes
14	Yes	Yes	Yes	XINT5	-	Yes	Yes	-	-	-	-	EPG1IN2	Yes	-	-	-	-	Yes	Yes
15	Yes	Yes	Yes	-	-	Yes	Yes	-	-	-	CLK1	EPG1IN3	Yes	-	-	-	-	Yes	Yes
16	Yes	Yes	Yes	-	-	Yes	Yes	-	-	-	CLK0	EPG1IN4	Yes	-	-	-	-	Yes	Yes
17	Yes	Yes	Yes	-	-	Yes	-	-	-	-	-	-	Yes	-	Yes	-	-	-	-
18	Yes	Yes	Yes	-	-	Yes	-	-	-	-	-	-	Yes	-	Yes	-	-	-	-
19	Yes	Yes	Yes	-	-	Yes	-	-	-	-	-	-	Yes	-	Yes	-	-	-	-
20	Yes	Yes	Yes	-	-	Yes	-	-	-	-	-	-	Yes	-	Yes	-	-	-	-
21	Yes	Yes	Yes	-	-	Yes	-	-	-	-	-	-	Yes	-	Yes	-	-	-	-
22	Yes	Yes	Yes	-	-	Yes	-	-	-	-	-	-	Yes	-	Yes	-	-	-	-
23	Yes	Yes	Yes	-	-	Yes	-	-	-	-	-	-	Yes	-	Yes	-	-	-	-
24	Yes	Yes	Yes	-	-	Yes	-	-	-	-	-	-	Yes	-	Yes	-	-	-	-
25	Yes	Yes	Yes	-	-	Yes	-	-	-	-	-	-	Yes	-	Yes	-	-	-	-
26	Yes	Yes	Yes	-	-	Yes	-	-	-	-	-	-	Yes	-	Yes	-	-	-	-
27	Yes	Yes	Yes	-	-	Yes	-	-	-	-	-	-	Yes	-	Yes	-	-	-	-
28	Yes	Yes	Yes	-	-	Yes	-	-	-	-	-	-	Yes	-	Yes	-	-	-	-
29	Yes	Yes	Yes	-	-	Yes	-	-	-	-	-	-	Yes	-	Yes	-	-	-	-
30	Yes	Yes	Yes	-	-	Yes	-	-	-	-	-	-	Yes	-	Yes	-	-	-	-
31	Yes	Yes	Yes	-	-	Yes	-	-	-	-	-	-	Yes	-	Yes	-	-	-	-
32	Yes	Yes	Yes	-	-	Yes	-	-	-	-	-	-	Yes	-	Yes	-	-	-	-
33	-	-	-	-	-	-	-	EQEP1A	-	-	-	-	-	-	-	WADI1	-	-	-
34	-	-	-	-	-	-	-	EQEP1B	-	-	-	-	-	-	-	WADI1	-	-	-
35	-	-	-	-	-	-	-	EQEP1I	-	-	-	-	-	-	-	WADI1	-	-	-
36	-	-	-	-	-	-	-	EQEP1S	-	-	-	-	-	-	-	WADI1	-	-	-

Table 20-1. Input X-BAR Destinations (continued)

INPUT	EPWMXBAR	CLBXBAR	OUTPUTXBAR	XINT	EPWM TRIP	EPWMDEL	ECAP	EQEP	ADCSOC	SYNCIN	DCCx	EPG	ERAD	CMPSS	CLB	WADI1	WADI2	ECAT	SENT
37	-	-	-	-	-	-	-	EQEP 2A	-	-	-	-	-	-	-	WADI1	-	-	-
38	-	-	-	-	-	-	-	EQEP 2B	-	-	-	-	-	-	-	WADI1	-	-	-
39	-	-	-	-	-	-	-	EQEP 2I	-	-	-	-	-	-	-	WADI1	-	-	-
40	-	-	-	-	-	-	-	EQEP 2S	-	-	-	-	-	-	-	WADI1	-	-	-
41	-	-	-	-	-	-	-	EQEP 3A	-	-	-	-	-	-	-	WADI1	-	-	-
42	-	-	-	-	-	-	-	EQEP 3B	-	-	-	-	-	-	-	WADI1	-	-	-
43	-	-	-	-	-	-	-	EQEP 3I	-	-	-	-	-	-	-	WADI1	-	-	-
44	-	-	-	-	-	-	-	EQEP 3S	-	-	-	-	-	-	-	WADI1	-	-	-
45	-	-	-	-	-	-	-	EQEP 4A	-	-	-	-	-	-	-	WADI1	-	-	-
46	-	-	-	-	-	-	-	EQEP 4B	-	-	-	-	-	-	-	WADI1	-	-	-
47	-	-	-	-	-	-	-	EQEP 4I	-	-	-	-	-	-	-	WADI1	-	-	-
48	-	-	-	-	-	-	-	EQEP 4S	-	-	-	-	-	-	-	WADI1	-	-	-
49	-	-	-	-	-	-	-	EQEP 5A	-	-	-	-	-	-	-	-	WADI2	-	-
50	-	-	-	-	-	-	-	EQEP 5B	-	-	-	-	-	-	-	-	WADI2	-	-
51	-	-	-	-	-	-	-	EQEP 5I	-	-	-	-	-	-	-	-	WADI2	-	-
52	-	-	-	-	-	-	-	EQEP 5S	-	-	-	-	-	-	-	-	WADI2	-	-
53	-	-	-	-	-	-	-	EQEP 6A	-	-	-	-	-	-	-	-	WADI2	-	-
54	-	-	-	-	-	-	-	EQEP 6B	-	-	-	-	-	-	-	-	WADI2	-	-

Table 20-1. Input X-BAR Destinations (continued)

INPUT	EPWMXBAR	CLBXBAR	OUTPUTXBAR	XINT	EPWM TRIP	EPWMDEL	ECAP	EQEP	ADCSOC	SYNCIN	DCCx	EPG	ERAD	CMPSS	CLB	WADI1	WADI2	ECAT	SENT
55	-	-	-	-	-	-	-	EQEP 6I	-	-	-	-	-	-	-	-	WADI2	-	-
56	-	-	-	-	-	-	-	EQEP 6S	-	-	-	-	-	-	-	-	WADI2	-	-
57	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	WADI2	-	-
58	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	WADI2	-	-
59	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	WADI2	-	-
60	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	WADI2	-	-
61	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	WADI2	-	-
62	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	WADI2	-	-
63	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	WADI2	-	-
64	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	WADI2	-	-

20.2.1 ICL and MINDB X-BAR

The Illegal Combo Logic (ICL) X-BAR and Minimum Dead-band (MINDB) X-BAR are used to route various EPWM signals and CLB outputs to the Minimum Dead-band and Illegal Combination Logic sub-module of the EPWM.The MINDB and ICL XBARs are architecturally equivalent to the CLB X-BAR and EPWM X-BAR. The inputs for each ICL X-BAR are listed in Table 20-2. The inputs for each MINDB X-BAR are listed in Table 20-3.

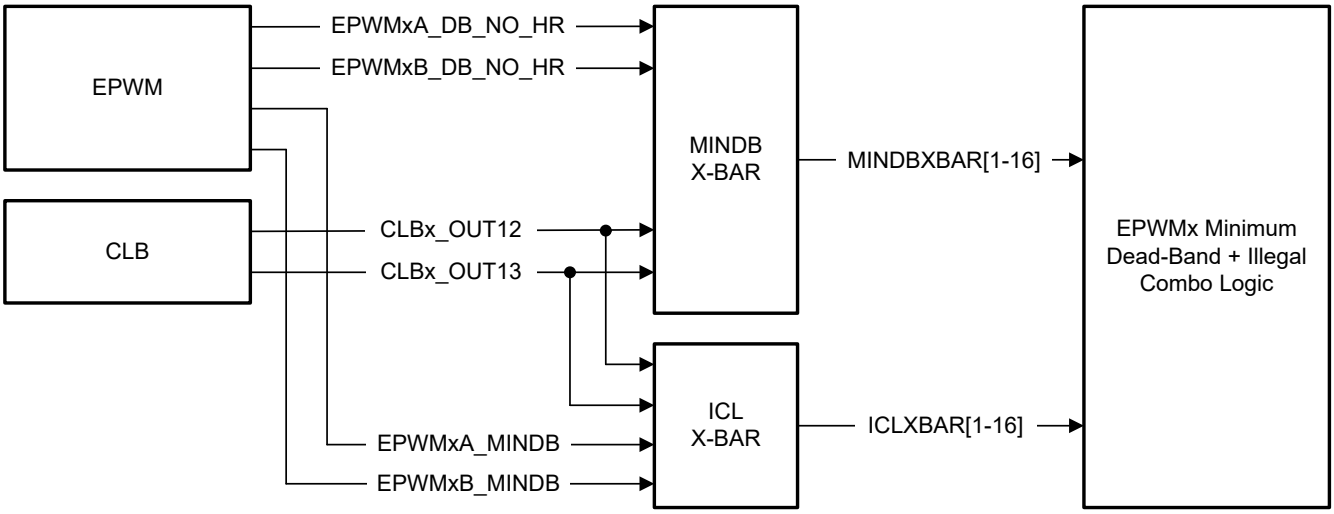


Figure 20-2. MINDB and ICL X-BAR

Table 20-2. Illegal Combination Logic X-BAR Mux Configuration Table

Group	Bit	Input Signal
G0	0	EPWM1A_MINDB
G0	1	EPWM2A_MINDB
G0	2	EPWM3A_MINDB
G0	3	EPWM4A_MINDB
G0	4	EPWM5A_MINDB
G0	5	EPWM6A_MINDB
G0	6	EPWM7A_MINDB
G0	7	EPWM8A_MINDB
G0	8	EPWM9A_MINDB
G0	9	EPWM10A_MINDB
G0	10	EPWM11A_MINDB
G0	11	EPWM12A_MINDB
G0	12	EPWM13A_MINDB
G0	13	EPWM14A_MINDB
G0	14	EPWM15A_MINDB
G0	15	EPWM16A_MINDB
G0	16	EPWM17A_MINDB
G0	17	EPWM18A_MINDB
G0	18	Reserved
G0	19	Reserved
G0	20	Reserved
G0	21	Reserved
G0	22	Reserved

Table 20-2. Illegal Combination Logic X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G0	23	Reserved
G0	24	Reserved
G0	25	Reserved
G0	26	Reserved
G0	27	Reserved
G0	28	Reserved
G0	29	Reserved
G0	30	Reserved
G0	31	Reserved
G1	0	EPWM1B_MINDB
G1	1	EPWM2B_MINDB
G1	2	EPWM3B_MINDB
G1	3	EPWM4B_MINDB
G1	4	EPWM5B_MINDB
G1	5	EPWM6B_MINDB
G1	6	EPWM7B_MINDB
G1	7	EPWM8B_MINDB
G1	8	EPWM9B_MINDB
G1	9	EPWM10B_MINDB
G1	10	EPWM11B_MINDB
G1	11	EPWM12B_MINDB
G1	12	EPWM13B_MINDB
G1	13	EPWM14B_MINDB
G1	14	EPWM15B_MINDB
G1	15	EPWM16B_MINDB
G1	16	EPWM17B_MINDB
G1	17	EPWM18B_MINDB
G1	18	Reserved
G1	19	Reserved
G1	20	Reserved
G1	21	Reserved
G1	22	Reserved
G1	23	Reserved
G1	24	Reserved
G1	25	Reserved
G1	26	Reserved
G1	27	Reserved
G1	28	Reserved
G1	29	Reserved
G1	30	Reserved
G1	31	Reserved
G2	0	CLB1_OUT12
G2	1	CLB1_OUT13
G2	2	CLB2_OUT12
G2	3	CLB2_OUT13

Table 20-2. Illegal Combination Logic X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G2	4	CLB3_OUT12
G2	5	CLB3_OUT13
G2	6	CLB4_OUT12
G2	7	CLB4_OUT13
G2	8	CLB5_OUT12
G2	9	CLB5_OUT13
G2	10	CLB6_OUT12
G2	11	CLB6_OUT13
G2	12	Reserved
G2	13	Reserved
G2	14	Reserved
G2	15	Reserved
G2	16	Reserved
G2	17	Reserved
G2	18	Reserved
G2	19	Reserved
G2	20	Reserved
G2	21	Reserved
G2	22	Reserved
G2	23	Reserved
G2	24	Reserved
G2	25	Reserved
G2	26	Reserved
G2	27	Reserved
G2	28	Reserved
G2	29	Reserved
G2	30	Reserved
G2	31	Reserved

Table 20-3. Minimum Deadband X-BAR Mux Configuration Table

Group	Bit	Input Signal
G0	0	EPWM1A_DB_NO_HR
G0	1	EPWM2A_DB_NO_HR
G0	2	EPWM3A_DB_NO_HR
G0	3	EPWM4A_DB_NO_HR
G0	4	EPWM5A_DB_NO_HR
G0	5	EPWM6A_DB_NO_HR
G0	6	EPWM7A_DB_NO_HR
G0	7	EPWM8A_DB_NO_HR
G0	8	EPWM9A_DB_NO_HR
G0	9	EPWM10A_DB_NO_HR
G0	10	EPWM11A_DB_NO_HR
G0	11	EPWM12A_DB_NO_HR
G0	12	EPWM13A_DB_NO_HR
G0	13	EPWM14A_DB_NO_HR

Table 20-3. Minimum Deadband X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G0	14	EPWM15A_DB_NO_HR
G0	15	EPWM16A_DB_NO_HR
G0	16	EPWM17A_DB_NO_HR
G0	17	EPWM18A_DB_NO_HR
G0	18	Reserved
G0	19	Reserved
G0	20	Reserved
G0	21	Reserved
G0	22	Reserved
G0	23	Reserved
G0	24	Reserved
G0	25	Reserved
G0	26	Reserved
G0	27	Reserved
G0	28	Reserved
G0	29	Reserved
G0	30	Reserved
G0	31	Reserved
G1	0	EPWM1B_DB_NO_HR
G1	1	EPWM2B_DB_NO_HR
G1	2	EPWM3B_DB_NO_HR
G1	3	EPWM4B_DB_NO_HR
G1	4	EPWM5B_DB_NO_HR
G1	5	EPWM6B_DB_NO_HR
G1	6	EPWM7B_DB_NO_HR
G1	7	EPWM8B_DB_NO_HR
G1	8	EPWM9B_DB_NO_HR
G1	9	EPWM10B_DB_NO_HR
G1	10	EPWM11B_DB_NO_HR
G1	11	EPWM12B_DB_NO_HR
G1	12	EPWM13B_DB_NO_HR
G1	13	EPWM14B_DB_NO_HR
G1	14	EPWM15B_DB_NO_HR
G1	15	EPWM16B_DB_NO_HR
G1	16	EPWM17B_DB_NO_HR
G1	17	EPWM18B_DB_NO_HR
G1	18	Reserved
G1	19	Reserved
G1	20	Reserved
G1	21	Reserved
G1	22	Reserved
G1	23	Reserved
G1	24	Reserved
G1	25	Reserved
G1	26	Reserved

Table 20-3. Minimum Deadband X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G1	27	Reserved
G1	28	Reserved
G1	29	Reserved
G1	30	Reserved
G1	31	Reserved
G2	0	CLB1_OUT12
G2	1	CLB1_OUT13
G2	2	CLB2_OUT12
G2	3	CLB2_OUT13
G2	4	CLB3_OUT12
G2	5	CLB3_OUT13
G2	6	CLB4_OUT12
G2	7	CLB4_OUT13
G2	8	CLB5_OUT12
G2	9	CLB5_OUT13
G2	10	CLB6_OUT12
G2	11	CLB6_OUT13
G2	12	Reserved
G2	13	Reserved
G2	14	Reserved
G2	15	Reserved
G2	16	Reserved
G2	17	Reserved
G2	18	Reserved
G2	19	Reserved
G2	20	Reserved
G2	21	Reserved
G2	22	Reserved
G2	23	Reserved
G2	24	Reserved
G2	25	Reserved
G2	26	Reserved
G2	27	Reserved
G2	28	Reserved
G2	29	Reserved
G2	30	Reserved
G2	31	Reserved

20.3 ePWM , CLB, and GPIO Output X-BAR

This section describes the ePWM , CLB, and GPIO Output X-BAR. Remember that the minimum input pulse width required for ePWM, CLB XBAR (CLB Clocks required), SYSCLK for Output XBAR is 3 ticks of the respective clocks.

20.3.1 ePWM X-BAR

The ePWM X-BAR brings signals to the ePWM modules. Specifically, the ePWM X-BAR is connected to the Digital Compare (DC) submodule of each ePWM module for actions such as tripzones and syncing. Refer to the *Enhanced Pulse Width Modulator (ePWM)* chapter for more information on additional ways the DC submodule can be used. Figure 20-3 shows the architecture of the ePWM X-BAR. Note that the architecture of the ePWM X-BAR is identical to the architecture of the GPIO Output X-BAR (with the exception of the output latch).

20.3.1.1 ePWM X-BAR Architecture

The ePWM X-BAR has eight outputs that are routed to each ePWM module. There are two instances of ePWM X-BAR (ePWM X-BAR A and ePWM X-BAR B), resulting in 16 outputs total. Figure 20-3 represents the architecture of a single output, but this output is identical to the architecture of all of the other outputs.

First, determine the signals that can be passed to the ePWM by referencing Table 20-4. Select up to one signal for each TRIPx output. Select the inputs to ePWM X-BAR using the PWMXBARGxSEL registers. To pass any signal through to the ePWM, enable the signal using the INPUTx bit in the PWMXBARGxSEL register. All signals that are enabled are logically ORed before being passed on to the respective TRIPx signal on the ePWM. To optionally invert the signal, use the PWMXBAROutInvert register.

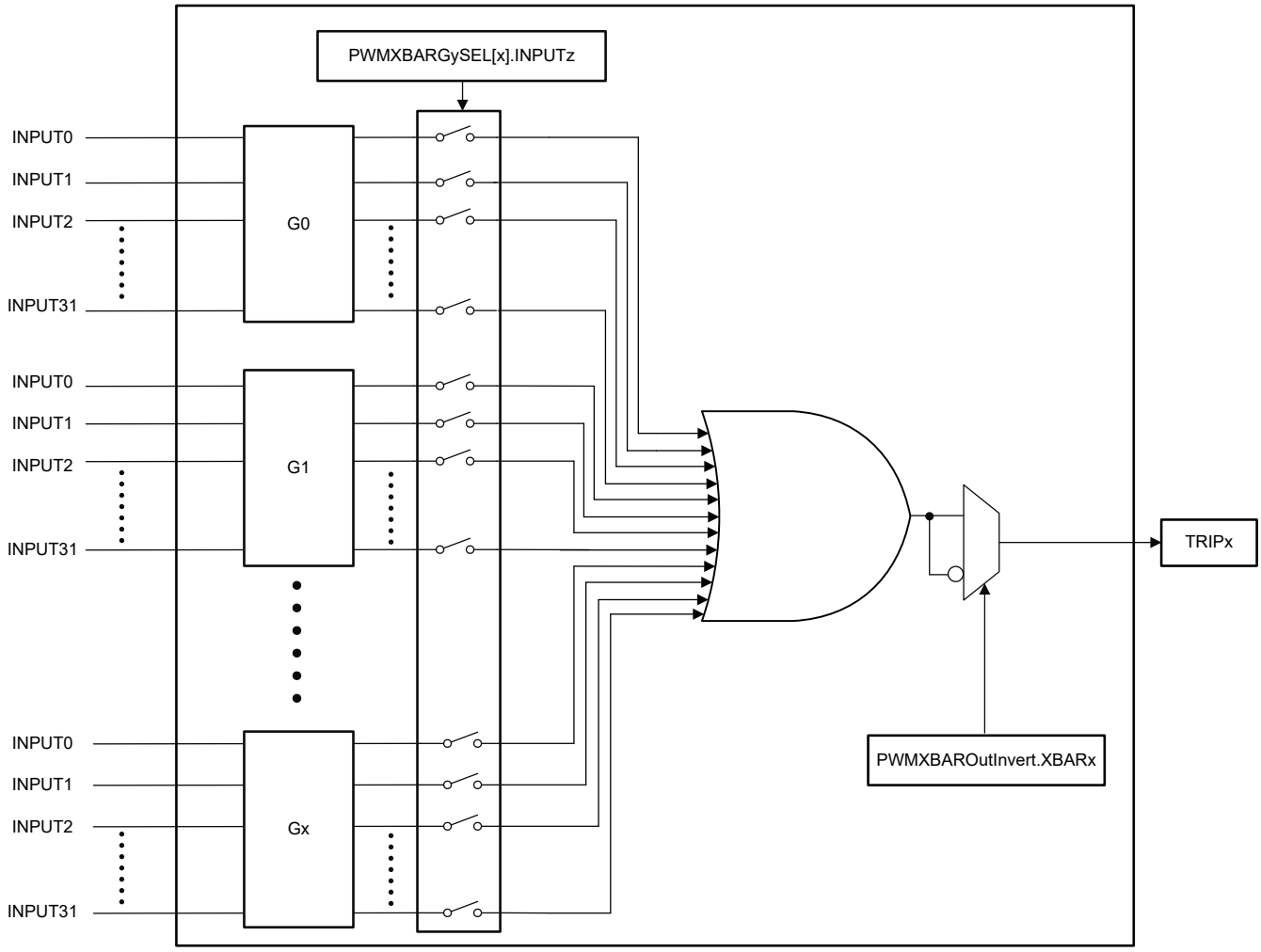


Figure 20-3. ePWM X-BAR Architecture - Single Output

Note

Do not use "Reserved" signals in your application.

There are two instances of the ePWM X-BAR (ePWM X-BAR A and ePWM X-BAR B). Table 20-4 describes the inputs to the ePWM X-BARs, which are identical for both instances. Each instance has 8 outputs. The outputs of ePWM X-BAR A are connected to trips 1-8 of each ePWM instance, while outputs 1-4 and output 6-7 of ePWM X-BAR B are connected to trips 9-12 and trips 14-15 of each ePWM instance, respectively. Note that outputs 5 and 8 of ePWM X-BAR B are not connected to any ePWM trip source.

Table 20-4. EPWM X-BAR Mux Configuration Table

Group	Bit	Input Signal
G0	0	CMPSS1_CTRIPH
G0	1	CMPSS1_CTRIPL
G0	2	CMPSS2_CTRIPH
G0	3	CMPSS2_CTRIPL
G0	4	CMPSS3_CTRIPH
G0	5	CMPSS3_CTRIPL
G0	6	CMPSS4_CTRIPH
G0	7	CMPSS4_CTRIPL
G0	8	CMPSS5_CTRIPH
G0	9	CMPSS5_CTRIPL
G0	10	CMPSS6_CTRIPH
G0	11	CMPSS6_CTRIPL
G0	12	CMPSS7_CTRIPH
G0	13	CMPSS7_CTRIPL
G0	14	CMPSS8_CTRIPH
G0	15	CMPSS8_CTRIPL
G0	16	CMPSS9_CTRIPH
G0	17	CMPSS9_CTRIPL
G0	18	CMPSS10_CTRIPH
G0	19	CMPSS10_CTRIPL
G0	20	CMPSS11_CTRIPH
G0	21	CMPSS11_CTRIPL
G0	22	CMPSS12_CTRIPH
G0	23	CMPSS12_CTRIPL
G0	24	Reserved
G0	25	Reserved
G0	26	Reserved
G0	27	Reserved
G0	28	Reserved
G0	29	Reserved
G0	30	Reserved

Table 20-4. EPWM X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G0	31	Reserved
G1	0	SD1FLT1_COMPH
G1	1	SD1FLT1_COMPL
G1	2	SD1FLT2_COMPH
G1	3	SD1FLT2_COMPL
G1	4	SD1FLT3_COMPH
G1	5	SD1FLT3_COMPL
G1	6	SD1FLT4_COMPH
G1	7	SD1FLT4_COMPL
G1	8	SD2FLT1_COMPH
G1	9	SD2FLT1_COMPL
G1	10	SD2FLT2_COMPH
G1	11	SD2FLT2_COMPL
G1	12	SD2FLT3_COMPH
G1	13	SD2FLT3_COMPL
G1	14	SD2FLT4_COMPH
G1	15	SD2FLT4_COMPL
G1	16	SD3FLT1_COMPH
G1	17	SD3FLT1_COMPL
G1	18	SD3FLT2_COMPH
G1	19	SD3FLT2_COMPL
G1	20	SD3FLT3_COMPH
G1	21	SD3FLT3_COMPL
G1	22	SD3FLT4_COMPH
G1	23	SD3FLT4_COMPL
G1	24	SD4FLT1_COMPH
G1	25	SD4FLT1_COMPL
G1	26	SD4FLT2_COMPH
G1	27	SD4FLT2_COMPL
G1	28	SD4FLT3_COMPH
G1	29	SD4FLT3_COMPL
G1	30	SD4FLT4_COMPH
G1	31	SD4FLT4_COMPL
G2	0	ADCAEVT1
G2	1	ADCAEVT2
G2	2	ADCAEVT3
G2	3	ADCAEVT4
G2	4	ADCBEVT1

Table 20-4. EPWM X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G2	5	ADCBEVT2
G2	6	ADCBEVT3
G2	7	ADCBEVT4
G2	8	ADCCEVT1
G2	9	ADCCEVT2
G2	10	ADCCEVT3
G2	11	ADCCEVT4
G2	12	ADCDEVT1
G2	13	ADCDEVT2
G2	14	ADCDEVT3
G2	15	ADCDEVT4
G2	16	ADCEEVT1
G2	17	ADCEEVT2
G2	18	ADCEEVT3
G2	19	ADCEEVT4
G2	20	CPU1_ADCCHECKEVT1
G2	21	CPU1_ADCCHECKEVT2
G2	22	CPU1_ADCCHECKEVT3
G2	23	CPU1_ADCCHECKEVT4
G2	24	CPU2_ADCCHECKEVT1
G2	25	CPU2_ADCCHECKEVT2
G2	26	CPU2_ADCCHECKEVT3
G2	27	CPU2_ADCCHECKEVT4
G2	28	CPU3_ADCCHECKEVT1
G2	29	CPU3_ADCCHECKEVT2
G2	30	CPU3_ADCCHECKEVT3
G2	31	CPU3_ADCCHECKEVT4
G3	0	INPUTXBAR1
G3	1	INPUTXBAR2
G3	2	INPUTXBAR3
G3	3	INPUTXBAR4
G3	4	INPUTXBAR5
G3	5	INPUTXBAR6
G3	6	INPUTXBAR7
G3	7	INPUTXBAR8
G3	8	INPUTXBAR9
G3	9	INPUTXBAR10
G3	10	INPUTXBAR11

Table 20-4. EPWM X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G3	11	INPUTXBAR12
G3	12	INPUTXBAR13
G3	13	INPUTXBAR14
G3	14	INPUTXBAR15
G3	15	INPUTXBAR16
G3	16	INPUTXBAR17
G3	17	INPUTXBAR18
G3	18	INPUTXBAR19
G3	19	INPUTXBAR20
G3	20	INPUTXBAR21
G3	21	INPUTXBAR22
G3	22	INPUTXBAR23
G3	23	INPUTXBAR24
G3	24	INPUTXBAR25
G3	25	INPUTXBAR26
G3	26	INPUTXBAR27
G3	27	INPUTXBAR28
G3	28	INPUTXBAR29
G3	29	INPUTXBAR30
G3	30	INPUTXBAR31
G3	31	INPUTXBAR32
G4	0	CLB1_OUT4
G4	1	CLB1_OUT5
G4	2	CLB2_OUT4
G4	3	CLB2_OUT5
G4	4	CLB3_OUT4
G4	5	CLB3_OUT5
G4	6	CLB4_OUT4
G4	7	CLB4_OUT5
G4	8	CLB5_OUT4
G4	9	CLB5_OUT5
G4	10	CLB6_OUT4
G4	11	CLB6_OUT5
G4	12	Reserved
G4	13	Reserved
G4	14	Reserved
G4	15	Reserved
G4	16	FSIRXA_TRIG1

Table 20-4. EPWM X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G4	17	FSIRXB_TRIG1
G4	18	FSIRXC_TRIG1
G4	19	FSIRXD_TRIG1
G4	20	Reserved
G4	21	Reserved
G4	22	Reserved
G4	23	Reserved
G4	24	Reserved
G4	25	Reserved
G4	26	Reserved
G4	27	Reserved
G4	28	Reserved
G4	29	Reserved
G4	30	ECAT_SYNC0
G4	31	ECAT_SYNC1
G5	0	ECAP1_OUT
G5	1	ECAP2_OUT
G5	2	ECAP3_OUT
G5	3	ECAP4_OUT
G5	4	ECAP5_OUT
G5	5	ECAP6_OUT
G5	6	Reserved
G5	7	Reserved
G5	8	ECAP1_TRIPOUT
G5	9	ECAP2_TRIPOUT
G5	10	ECAP3_TRIPOUT
G5	11	ECAP4_TRIPOUT
G5	12	ECAP5_TRIPOUT
G5	13	ECAP6_TRIPOUT
G5	14	Reserved
G5	15	Reserved
G5	16	ADCSOCA
G5	17	ADCSOCB
G5	18	ESM_GEN_EVENT
G5	19	EXTSYNCOUT
G5	20	WADI1_ERROR_BUS_O[0]
G5	21	WADI1_ERROR_BUS_O[1]
G5	22	WADI1_ERROR_BUS_O[2]

Table 20-4. EPWM X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G5	23	WADI1_ERROR_BUS_O[3]
G5	24	WADI1_ERROR_BUS_O[4]
G5	25	WADI1_ERROR_BUS_O[5]
G5	26	WADI1_ERROR_BUS_O[6]
G5	27	WADI1_ERROR_BUS_O[7]
G5	28	WADI1_ERROR_BUS_O[8]
G5	29	WADI1_ERROR_BUS_O[9]
G5	30	WADI1_ERROR_BUS_O[10]
G5	31	WADI1_ERROR_BUS_O[11]
G6	0	MCANA_FEVT0
G6	1	MCANA_FEVT1
G6	2	MCANA_FEVT2
G6	3	MCANB_FEVT0
G6	4	MCANB_FEVT1
G6	5	MCANB_FEVT2
G6	6	MCANC_FEVT0
G6	7	MCANC_FEVT1
G6	8	MCANC_FEVT2
G6	9	MCAND_FEVT0
G6	10	MCAND_FEVT1
G6	11	MCAND_FEVT2
G6	12	MCANE_FEVT0
G6	13	MCANE_FEVT1
G6	14	MCANE_FEVT2
G6	15	MCANF_FEVT0
G6	16	MCANF_FEVT1
G6	17	MCANF_FEVT2
G6	18	Reserved
G6	19	Reserved
G6	20	WADI2_ERROR_BUS_O[0]
G6	21	WADI2_ERROR_BUS_O[1]
G6	22	WADI2_ERROR_BUS_O[2]
G6	23	WADI2_ERROR_BUS_O[3]
G6	24	WADI2_ERROR_BUS_O[4]
G6	25	WADI2_ERROR_BUS_O[5]
G6	26	WADI2_ERROR_BUS_O[6]
G6	27	WADI2_ERROR_BUS_O[7]
G6	28	WADI2_ERROR_BUS_O[8]

Table 20-4. EPWM X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G6	29	WADI2_ERROR_BUS_O[9]
G6	30	WADI2_ERROR_BUS_O[10]
G6	31	WADI2_ERROR_BUS_O[11]
G7	0	EPWM1_TRIPOUT
G7	1	EPWM2_TRIPOUT
G7	2	EPWM3_TRIPOUT
G7	3	EPWM4_TRIPOUT
G7	4	EPWM5_TRIPOUT
G7	5	EPWM6_TRIPOUT
G7	6	EPWM7_TRIPOUT
G7	7	EPWM8_TRIPOUT
G7	8	EPWM9_TRIPOUT
G7	9	EPWM10_TRIPOUT
G7	10	EPWM11_TRIPOUT
G7	11	EPWM12_TRIPOUT
G7	12	EPWM13_TRIPOUT
G7	13	EPWM14_TRIPOUT
G7	14	EPWM15_TRIPOUT
G7	15	EPWM16_TRIPOUT
G7	16	EPWM17_TRIPOUT
G7	17	EPWM18_TRIPOUT
G7	18	Reserved
G7	19	Reserved
G7	20	Reserved
G7	21	Reserved
G7	22	Reserved
G7	23	Reserved
G7	24	Reserved
G7	25	Reserved
G7	26	Reserved
G7	27	Reserved
G7	28	Reserved
G7	29	Reserved
G7	30	Reserved
G7	31	Reserved
G8	0	EPWM1_DE_DEMONTRIP
G8	1	EPWM2_DE_DEMONTRIP
G8	2	EPWM3_DE_DEMONTRIP

Table 20-4. EPWM X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G8	3	EPWM4_DE_DEMONTRIP
G8	4	EPWM5_DE_DEMONTRIP
G8	5	EPWM6_DE_DEMONTRIP
G8	6	EPWM7_DE_DEMONTRIP
G8	7	EPWM8_DE_DEMONTRIP
G8	8	EPWM9_DE_DEMONTRIP
G8	9	EPWM10_DE_DEMONTRIP
G8	10	EPWM11_DE_DEMONTRIP
G8	11	EPWM12_DE_DEMONTRIP
G8	12	EPWM13_DE_DEMONTRIP
G8	13	EPWM14_DE_DEMONTRIP
G8	14	EPWM15_DE_DEMONTRIP
G8	15	EPWM16_DE_DEMONTRIP
G8	16	EPWM17_DE_DEMONTRIP
G8	17	EPWM18_DE_DEMONTRIP
G8	18	Reserved
G8	19	Reserved
G8	20	Reserved
G8	21	Reserved
G8	22	Reserved
G8	23	Reserved
G8	24	Reserved
G8	25	Reserved
G8	26	Reserved
G8	27	Reserved
G8	28	Reserved
G8	29	Reserved
G8	30	Reserved
G8	31	Reserved
G9	0	EPWM1_DE_ACTIVE
G9	1	EPWM2_DE_ACTIVE
G9	2	EPWM3_DE_ACTIVE
G9	3	EPWM4_DE_ACTIVE
G9	4	EPWM5_DE_ACTIVE
G9	5	EPWM6_DE_ACTIVE
G9	6	EPWM7_DE_ACTIVE
G9	7	EPWM8_DE_ACTIVE
G9	8	EPWM9_DE_ACTIVE

Table 20-4. EPWM X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G9	9	EPWM10_DE_ACTIVE
G9	10	EPWM11_DE_ACTIVE
G9	11	EPWM12_DE_ACTIVE
G9	12	EPWM13_DE_ACTIVE
G9	13	EPWM14_DE_ACTIVE
G9	14	EPWM15_DE_ACTIVE
G9	15	EPWM16_DE_ACTIVE
G9	16	EPWM17_DE_ACTIVE
G9	17	EPWM18_DE_ACTIVE
G9	18	Reserved
G9	19	Reserved
G9	20	Reserved
G9	21	Reserved
G9	22	Reserved
G9	23	Reserved
G9	24	Reserved
G9	25	Reserved
G9	26	Reserved
G9	27	Reserved
G9	28	Reserved
G9	29	Reserved
G9	30	Reserved
G9	31	Reserved

20.3.2 CLB X-BAR

The CLB X-BAR brings signals to the CLB modules. Figure 20-4 shows the architecture of the CLB X-BAR.

20.3.2.1 CLB X-BAR Architecture

The CLB X-BAR has eight outputs that are routed to each CLB module. Figure 20-4 represents the architecture of a single output, but the output is identical to the architecture of all of the other outputs.

First, determine the signals that can be passed to the CLB by referencing Table 20-5. Each of these signals are enabled or disabled using the CLBXBARxGySEL registers before being passed through an OR gate. Additionally, there is an option to invert the output of the OR gate before the signal is passed to one of the auxiliary inputs (AUXSIG) of the CLB. There is one instance of CLB-XBAR per AUXSIG input. This architecture is identical to the ePWM X-BAR, ICL and MINDB X-BARs.

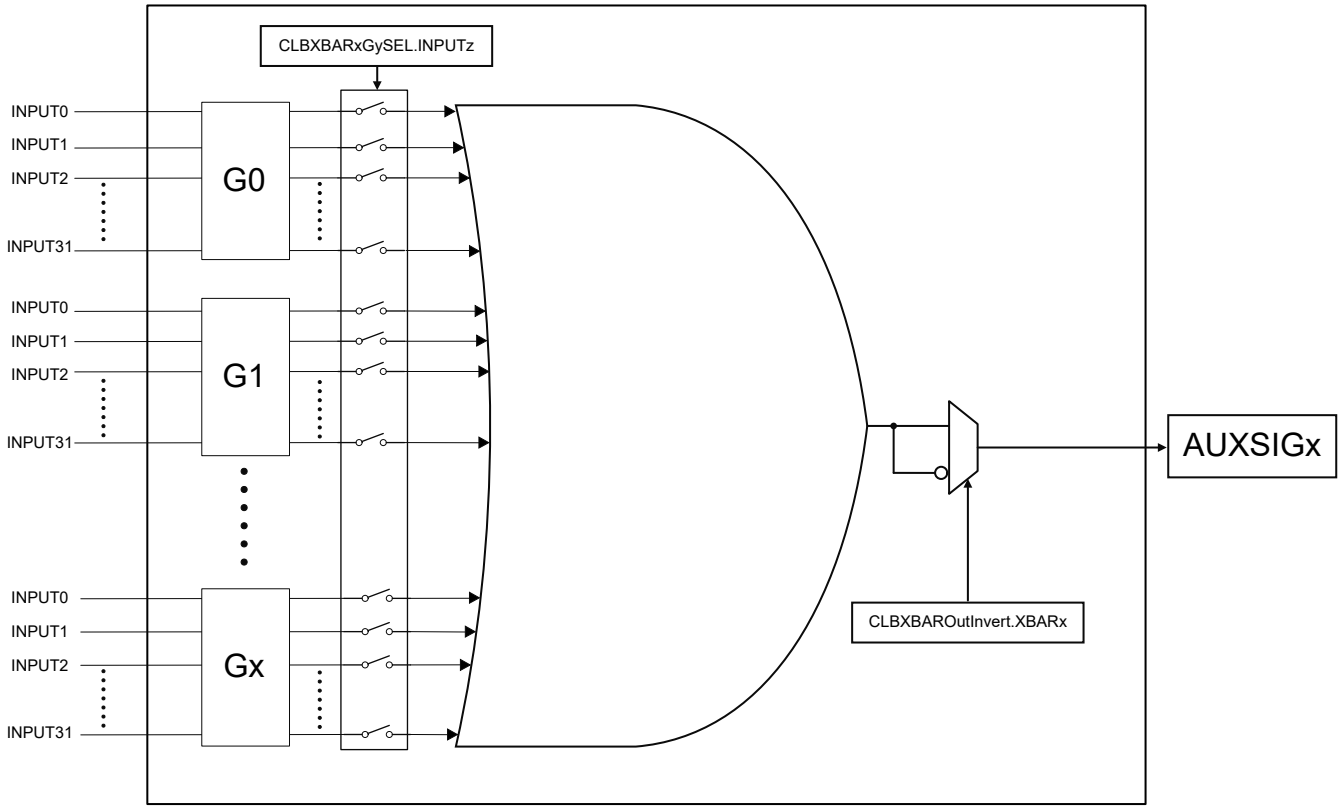


Figure 20-4. CLB X-BAR Architecture - Single Output

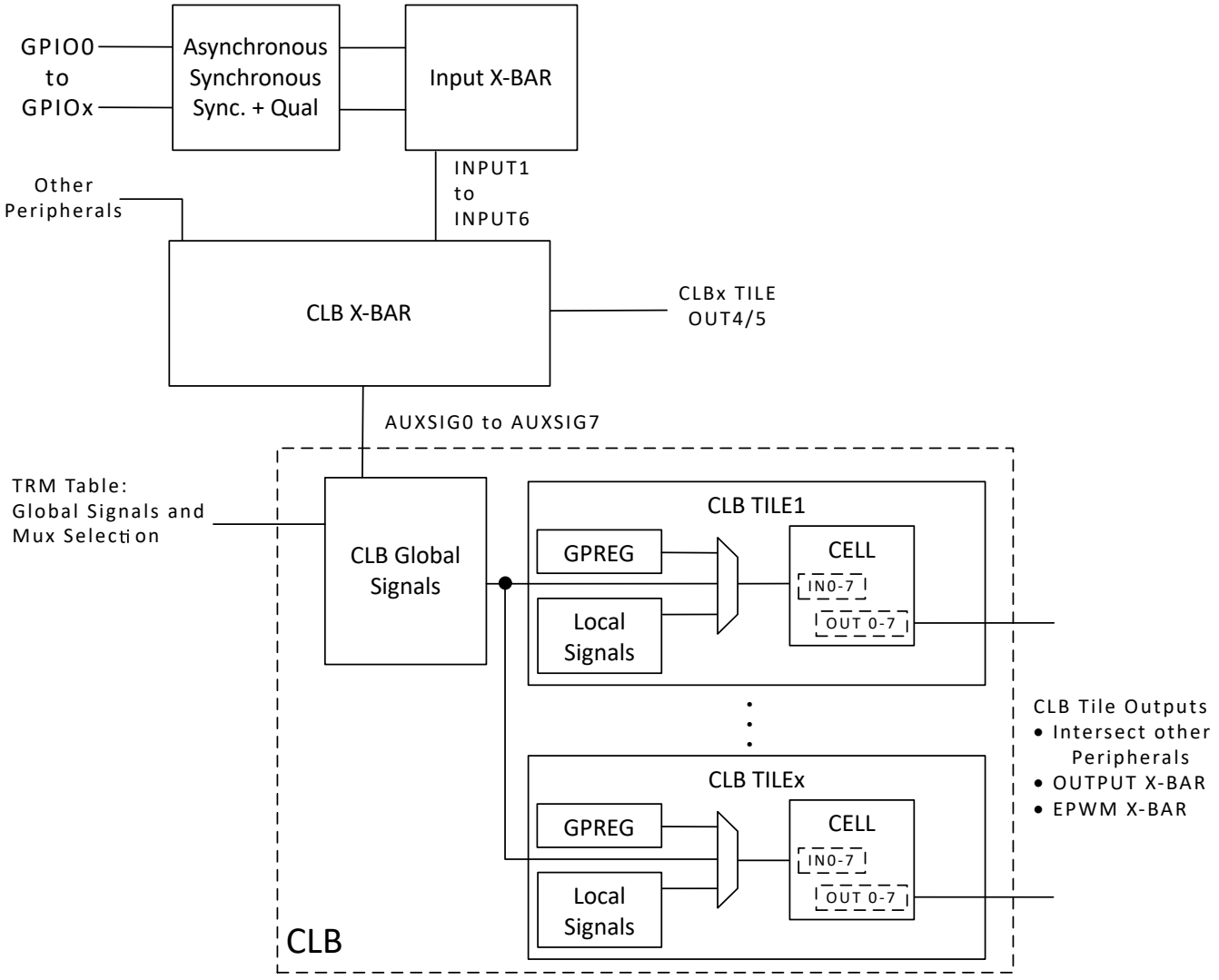


Figure 20-5. GPIO to CLB Tile Connections

Table 20-5. CLB X-BAR Mux Configuration Table

Group	Bit	Input Signal
G0	0	CMPSS1_CTRIPH
G0	1	CMPSS1_CTRIPL
G0	2	CMPSS2_CTRIPH
G0	3	CMPSS2_CTRIPL
G0	4	CMPSS3_CTRIPH
G0	5	CMPSS3_CTRIPL
G0	6	CMPSS4_CTRIPH
G0	7	CMPSS4_CTRIPL
G0	8	CMPSS5_CTRIPH
G0	9	CMPSS5_CTRIPL
G0	10	CMPSS6_CTRIPH

Table 20-5. CLB X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G0	11	CMPSS6_CTRIPL
G0	12	CMPSS7_CTRIPH
G0	13	CMPSS7_CTRIPL
G0	14	CMPSS8_CTRIPH
G0	15	CMPSS8_CTRIPL
G0	16	CMPSS9_CTRIPH
G0	17	CMPSS9_CTRIPL
G0	18	CMPSS10_CTRIPH
G0	19	CMPSS10_CTRIPL
G0	20	CMPSS11_CTRIPH
G0	21	CMPSS11_CTRIPL
G0	22	CMPSS12_CTRIPH
G0	23	CMPSS12_CTRIPL
G0	24	Reserved
G0	25	Reserved
G0	26	Reserved
G0	27	Reserved
G0	28	Reserved
G0	29	Reserved
G0	30	Reserved
G0	31	Reserved
G1	0	SD1FLT1_COMPH
G1	1	SD1FLT1_COMPL
G1	2	SD1FLT2_COMPH
G1	3	SD1FLT2_COMPL
G1	4	SD1FLT3_COMPH
G1	5	SD1FLT3_COMPL
G1	6	SD1FLT4_COMPH
G1	7	SD1FLT4_COMPL
G1	8	SD2FLT1_COMPH
G1	9	SD2FLT1_COMPL
G1	10	SD2FLT2_COMPH
G1	11	SD2FLT2_COMPL
G1	12	SD2FLT3_COMPH
G1	13	SD2FLT3_COMPL
G1	14	SD2FLT4_COMPH
G1	15	SD2FLT4_COMPL
G1	16	SD3FLT1_COMPH

Table 20-5. CLB X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G1	17	SD3FLT1_COMPL
G1	18	SD3FLT2_COMPH
G1	19	SD3FLT2_COMPL
G1	20	SD3FLT3_COMPH
G1	21	SD3FLT3_COMPL
G1	22	SD3FLT4_COMPH
G1	23	SD3FLT4_COMPL
G1	24	SD4FLT1_COMPH
G1	25	SD4FLT1_COMPL
G1	26	SD4FLT2_COMPH
G1	27	SD4FLT2_COMPL
G1	28	SD4FLT3_COMPH
G1	29	SD4FLT3_COMPL
G1	30	SD4FLT4_COMPH
G1	31	SD4FLT4_COMPL
G2	0	ADCAEVT1
G2	1	ADCAEVT2
G2	2	ADCAEVT3
G2	3	ADCAEVT4
G2	4	ADCBEVT1
G2	5	ADCBEVT2
G2	6	ADCBEVT3
G2	7	ADCBEVT4
G2	8	ADCCEVT1
G2	9	ADCCEVT2
G2	10	ADCCEVT3
G2	11	ADCCEVT4
G2	12	ADCDEVT1
G2	13	ADCDEVT2
G2	14	ADCDEVT3
G2	15	ADCDEVT4
G2	16	ADCEEVT1
G2	17	ADCEEVT2
G2	18	ADCEEVT3
G2	19	ADCEEVT4
G2	20	CPU1_ADCCHECKEVT1
G2	21	CPU1_ADCCHECKEVT2
G2	22	CPU1_ADCCHECKEVT3

Table 20-5. CLB X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G2	23	CPU1_ADCCHECKEVT4
G2	24	CPU2_ADCCHECKEVT1
G2	25	CPU2_ADCCHECKEVT2
G2	26	CPU2_ADCCHECKEVT3
G2	27	CPU2_ADCCHECKEVT4
G2	28	CPU3_ADCCHECKEVT1
G2	29	CPU3_ADCCHECKEVT2
G2	30	CPU3_ADCCHECKEVT3
G2	31	CPU3_ADCCHECKEVT4
G3	0	INPUTXBAR1
G3	1	INPUTXBAR2
G3	2	INPUTXBAR3
G3	3	INPUTXBAR4
G3	4	INPUTXBAR5
G3	5	INPUTXBAR6
G3	6	INPUTXBAR7
G3	7	INPUTXBAR8
G3	8	INPUTXBAR9
G3	9	INPUTXBAR10
G3	10	INPUTXBAR11
G3	11	INPUTXBAR12
G3	12	INPUTXBAR13
G3	13	INPUTXBAR14
G3	14	INPUTXBAR15
G3	15	INPUTXBAR16
G3	16	INPUTXBAR17
G3	17	INPUTXBAR18
G3	18	INPUTXBAR19
G3	19	INPUTXBAR20
G3	20	INPUTXBAR21
G3	21	INPUTXBAR22
G3	22	INPUTXBAR23
G3	23	INPUTXBAR24
G3	24	INPUTXBAR25
G3	25	INPUTXBAR26
G3	26	INPUTXBAR27
G3	27	INPUTXBAR28
G3	28	INPUTXBAR29

Table 20-5. CLB X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G3	29	INPUTXBAR30
G3	30	INPUTXBAR31
G3	31	INPUTXBAR32
G4	0	CLB1_OUT12
G4	1	CLB1_OUT13
G4	2	CLB2_OUT12
G4	3	CLB2_OUT13
G4	4	CLB3_OUT12
G4	5	CLB3_OUT13
G4	6	CLB4_OUT12
G4	7	CLB4_OUT13
G4	8	CLB5_OUT12
G4	9	CLB5_OUT13
G4	10	CLB6_OUT12
G4	11	CLB6_OUT13
G4	12	Reserved
G4	13	Reserved
G4	14	Reserved
G4	15	Reserved
G4	16	FSIRXA_TRIG1
G4	17	FSIRXB_TRIG1
G4	18	FSIRXC_TRIG1
G4	19	FSIRXD_TRIG1
G4	20	FSIRXA_TRIG2
G4	21	FSIRXB_TRIG2
G4	22	FSIRXC_TRIG2
G4	23	FSIRXD_TRIG2
G4	24	FSIRXA_TRIG3
G4	25	FSIRXB_TRIG3
G4	26	FSIRXC_TRIG3
G4	27	FSIRXD_TRIG3
G4	28	Reserved
G4	29	Reserved
G4	30	ECAT_SYNC0
G4	31	ECAT_SYNC1
G5	0	ECAP1_OUT
G5	1	ECAP2_OUT
G5	2	ECAP3_OUT

Table 20-5. CLB X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G5	3	ECAP4_OUT
G5	4	ECAP5_OUT
G5	5	ECAP6_OUT
G5	6	Reserved
G5	7	Reserved
G5	8	ECAP1_TRIPOUT
G5	9	ECAP2_TRIPOUT
G5	10	ECAP3_TRIPOUT
G5	11	ECAP4_TRIPOUT
G5	12	ECAP5_TRIPOUT
G5	13	ECAP6_TRIPOUT
G5	14	Reserved
G5	15	Reserved
G5	16	ADCSOCA
G5	17	ADCSOCB
G5	18	ESM_GEN_EVENT
G5	19	EXTSYNCOUT
G5	20	Reserved
G5	21	Reserved
G5	22	Reserved
G5	23	Reserved
G5	24	Reserved
G5	25	Reserved
G5	26	Reserved
G5	27	Reserved
G5	28	Reserved
G5	29	Reserved
G5	30	Reserved
G5	31	Reserved
G6	0	MCANA_FEVT0
G6	1	MCANA_FEVT1
G6	2	MCANA_FEVT2
G6	3	MCANB_FEVT0
G6	4	MCANB_FEVT1
G6	5	MCANB_FEVT2
G6	6	MCANC_FEVT0
G6	7	MCANC_FEVT1
G6	8	MCANC_FEVT2

Table 20-5. CLB X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G6	9	MCAND_FEVT0
G6	10	MCAND_FEVT1
G6	11	MCAND_FEVT2
G6	12	MCANE_FEVT0
G6	13	MCANE_FEVT1
G6	14	MCANE_FEVT2
G6	15	MCANF_FEVT0
G6	16	MCANF_FEVT1
G6	17	MCANF_FEVT2
G6	18	Reserved
G6	19	Reserved
G6	20	CPU1_ERAD_EVT8
G6	21	CPU1_ERAD_EVT9
G6	22	CPU1_ERAD_EVT10
G6	23	CPU1_ERAD_EVT11
G6	24	CPU2_ERAD_EVT8
G6	25	CPU2_ERAD_EVT9
G6	26	CPU2_ERAD_EVT10
G6	27	CPU2_ERAD_EVT11
G6	28	CPU3_ERAD_EVT8
G6	29	CPU3_ERAD_EVT9
G6	30	CPU3_ERAD_EVT10
G6	31	CPU3_ERAD_EVT11
G7	0	WADI1OUT0
G7	1	WADI1OUT1
G7	2	WADI1OUT2
G7	3	WADI1OUT3
G7	4	WADI1OUT4
G7	5	WADI1OUT5
G7	6	WADI1OUT6
G7	7	WADI1OUT7
G7	8	WADI2OUT0
G7	9	WADI2OUT1
G7	10	WADI2OUT2
G7	11	WADI2OUT3
G7	12	WADI2OUT4
G7	13	WADI2OUT5
G7	14	WADI2OUT6

Table 20-5. CLB X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G7	15	WADI2OUT7
G7	16	Reserved
G7	17	Reserved
G7	18	Reserved
G7	19	Reserved
G7	20	Reserved
G7	21	Reserved
G7	22	Reserved
G7	23	Reserved
G7	24	Reserved
G7	25	Reserved
G7	26	Reserved
G7	27	Reserved
G7	28	Reserved
G7	29	Reserved
G7	30	Reserved
G7	31	Reserved

20.3.3 GPIO Output X-BAR

The GPIO Output X-BAR takes signals from inside the device and brings them out to a GPIO. Figure 20-6 shows the architecture of the GPIO Output X-BAR. The X-BAR contains sixteen outputs and each contains at least one position on the GPIO mux, denoted as OUTPUTXBARx. The X-BAR allows the selection of a single input or a logical-OR of many inputs.

20.3.3.1 GPIO Output X-BAR Architecture

The GPIO Output X-BAR has sixteen outputs that are routed to the GPIO module. Figure 20-6 represents the architecture of a single output, but this output is identical to the architecture of all of the other outputs.

First, determine the signals that can be passed to the GPIO by referencing Table 20-6. Each of these signals are enabled or disabled via the OUTPUTBARGxSEL registers before being passed through an OR gate. In contrast to the other XBAR modules, there are also features for output stretching, latching the output, and inverting the output either at the flag or final output stage. The diagram below shows these features in depth. The final output is only recognized on the GPIO if the proper OUTPUTx muxing options are selected using the GPIO registers.

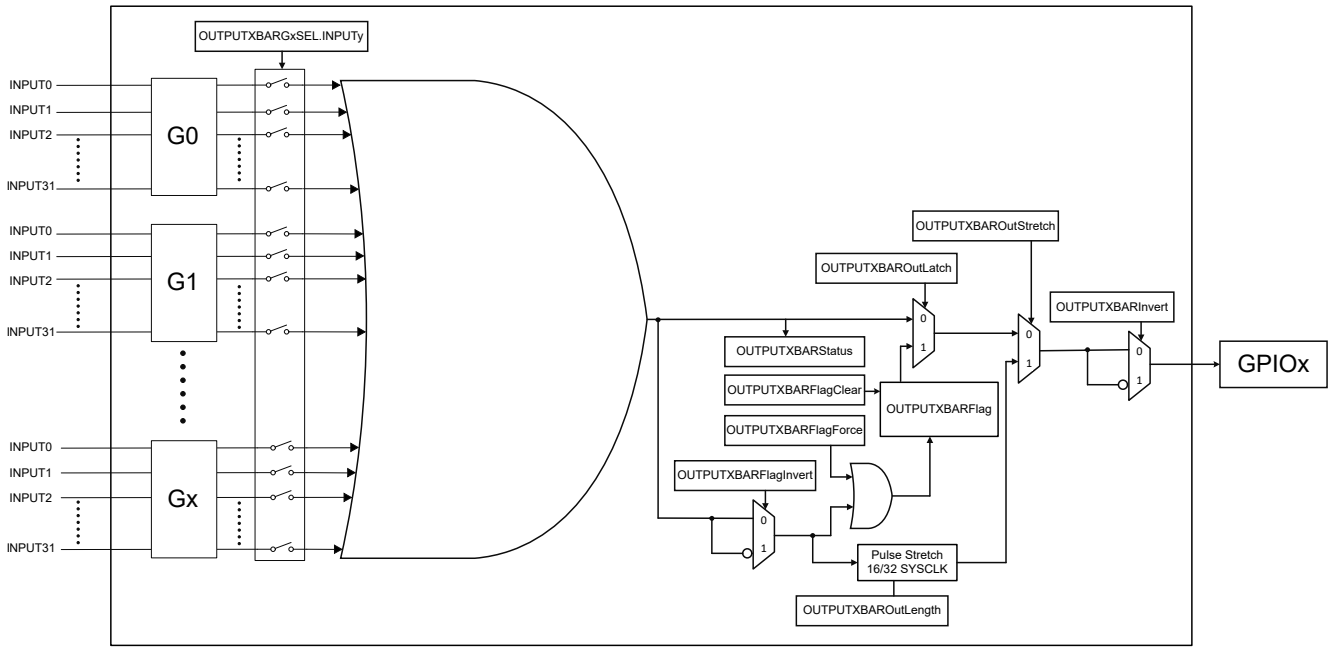


Figure 20-6. GPIO Output X-BAR Architecture

Note

Do not use "Reserved" signals in your application.

Table 20-6. Output X-BAR Mux Configuration Table

Group	Bit	Input Signal
G0	0	CMPSS1_CTRIPOUTH
G0	1	CMPSS1_CTRIPOUTL
G0	2	CMPSS2_CTRIPOUTH
G0	3	CMPSS2_CTRIPOUTL
G0	4	CMPSS3_CTRIPOUTH
G0	5	CMPSS3_CTRIPOUTL
G0	6	CMPSS4_CTRIPOUTH

Table 20-6. Output X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G0	7	CMPSS4_CTRIPOUTL
G0	8	CMPSS5_CTRIPOUTH
G0	9	CMPSS5_CTRIPOUTL
G0	10	CMPSS6_CTRIPOUTH
G0	11	CMPSS6_CTRIPOUTL
G0	12	CMPSS7_CTRIPOUTH
G0	13	CMPSS7_CTRIPOUTL
G0	14	CMPSS8_CTRIPOUTH
G0	15	CMPSS8_CTRIPOUTL
G0	16	CMPSS9_CTRIPOUTH
G0	17	CMPSS9_CTRIPOUTL
G0	18	CMPSS10_CTRIPOUTH
G0	19	CMPSS10_CTRIPOUTL
G0	20	CMPSS11_CTRIPOUTH
G0	21	CMPSS11_CTRIPOUTL
G0	22	CMPSS12_CTRIPOUTH
G0	23	CMPSS12_CTRIPOUTL
G0	24	Reserved
G0	25	Reserved
G0	26	Reserved
G0	27	Reserved
G0	28	Reserved
G0	29	Reserved
G0	30	Reserved
G0	31	Reserved
G1	0	SD1FLT1_COMPH
G1	1	SD1FLT1_COMPL
G1	2	SD1FLT2_COMPH
G1	3	SD1FLT2_COMPL
G1	4	SD1FLT3_COMPH
G1	5	SD1FLT3_COMPL
G1	6	SD1FLT4_COMPH
G1	7	SD1FLT4_COMPL
G1	8	SD2FLT1_COMPH
G1	9	SD2FLT1_COMPL
G1	10	SD2FLT2_COMPH
G1	11	SD2FLT2_COMPL
G1	12	SD2FLT3_COMPH

Table 20-6. Output X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G1	13	SD2FLT3_COMPL
G1	14	SD2FLT4_COMPH
G1	15	SD2FLT4_COMPL
G1	16	SD3FLT1_COMPH
G1	17	SD3FLT1_COMPL
G1	18	SD3FLT2_COMPH
G1	19	SD3FLT2_COMPL
G1	20	SD3FLT3_COMPH
G1	21	SD3FLT3_COMPL
G1	22	SD3FLT4_COMPH
G1	23	SD3FLT4_COMPL
G1	24	SD4FLT1_COMPH
G1	25	SD4FLT1_COMPL
G1	26	SD4FLT2_COMPH
G1	27	SD4FLT2_COMPL
G1	28	SD4FLT3_COMPH
G1	29	SD4FLT3_COMPL
G1	30	SD4FLT4_COMPH
G1	31	SD4FLT4_COMPL
G2	0	ADCAEVT1
G2	1	ADCAEVT2
G2	2	ADCAEVT3
G2	3	ADCAEVT4
G2	4	ADCBEVT1
G2	5	ADCBEVT2
G2	6	ADCBEVT3
G2	7	ADCBEVT4
G2	8	ADCCEVT1
G2	9	ADCCEVT2
G2	10	ADCCEVT3
G2	11	ADCCEVT4
G2	12	ADCDEVT1
G2	13	ADCDEVT2
G2	14	ADCDEVT3
G2	15	ADCDEVT4
G2	16	ADCEEVT1
G2	17	ADCEEVT2
G2	18	ADCEEVT3

Table 20-6. Output X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G2	19	ADCEEVT4
G2	20	CPU1_ADCCHECKEVT1
G2	21	CPU1_ADCCHECKEVT2
G2	22	CPU1_ADCCHECKEVT3
G2	23	CPU1_ADCCHECKEVT4
G2	24	CPU2_ADCCHECKEVT1
G2	25	CPU2_ADCCHECKEVT2
G2	26	CPU2_ADCCHECKEVT3
G2	27	CPU2_ADCCHECKEVT4
G2	28	CPU3_ADCCHECKEVT1
G2	29	CPU3_ADCCHECKEVT2
G2	30	CPU3_ADCCHECKEVT3
G2	31	CPU3_ADCCHECKEVT4
G3	0	INPUTXBAR1
G3	1	INPUTXBAR2
G3	2	INPUTXBAR3
G3	3	INPUTXBAR4
G3	4	INPUTXBAR5
G3	5	INPUTXBAR6
G3	6	INPUTXBAR7
G3	7	INPUTXBAR8
G3	8	Reserved
G3	9	Reserved
G3	10	Reserved
G3	11	Reserved
G3	12	Reserved
G3	13	Reserved
G3	14	Reserved
G3	15	Reserved
G3	16	Reserved
G3	17	Reserved
G3	18	Reserved
G3	19	Reserved
G3	20	Reserved
G3	21	Reserved
G3	22	Reserved
G3	23	Reserved
G3	24	Reserved

Table 20-6. Output X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G3	25	Reserved
G3	26	Reserved
G3	27	Reserved
G3	28	Reserved
G3	29	Reserved
G3	30	Reserved
G3	31	Reserved
G4	0	EQEP1_INDEX_SYNCOUT
G4	1	EQEP1_STROBE_SYNCOUT
G4	2	EQEP2_INDEX_SYNCOUT
G4	3	EQEP2_STROBE_SYNCOUT
G4	4	EQEP3_INDEX_SYNCOUT
G4	5	EQEP3_STROBE_SYNCOUT
G4	6	EQEP4_INDEX_SYNCOUT
G4	7	EQEP4_STROBE_SYNCOUT
G4	8	EQEP5_INDEX_SYNCOUT
G4	9	EQEP5_STROBE_SYNCOUT
G4	10	EQEP6_INDEX_SYNCOUT
G4	11	EQEP6_STROBE_SYNCOUT
G4	12	Reserved
G4	13	Reserved
G4	14	Reserved
G4	15	Reserved
G4	16	FSIRXA_TRIG1
G4	17	FSIRXB_TRIG1
G4	18	FSIRXC_TRIG1
G4	19	FSIRXD_TRIG1
G4	20	FSIRXA_TRIG2
G4	21	FSIRXB_TRIG2
G4	22	FSIRXC_TRIG2
G4	23	FSIRXD_TRIG2
G4	24	FSIRXA_TRIG3
G4	25	FSIRXB_TRIG3
G4	26	FSIRXC_TRIG3
G4	27	FSIRXD_TRIG3
G4	28	Reserved
G4	29	XCLKOUT
G4	30	ECAT_SYNC0

Table 20-6. Output X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G4	31	ECAT_SYNC1
G5	0	ECAP1_OUT
G5	1	ECAP2_OUT
G5	2	ECAP3_OUT
G5	3	ECAP4_OUT
G5	4	ECAP5_OUT
G5	5	ECAP6_OUT
G5	6	Reserved
G5	7	Reserved
G5	8	ECAP1_TRIPOUT
G5	9	ECAP2_TRIPOUT
G5	10	ECAP3_TRIPOUT
G5	11	ECAP4_TRIPOUT
G5	12	ECAP5_TRIPOUT
G5	13	ECAP6_TRIPOUT
G5	14	Reserved
G5	15	Reserved
G5	16	ADCSOCA
G5	17	ADCSOCB
G5	18	ESM_GEN_EVENT
G5	19	EXTSYNCOUT
G5	20	EPG1OUT0
G5	21	EPG1OUT1
G5	22	EPG1OUT2
G5	23	EPG1OUT3
G5	24	WADI1OUT0
G5	25	WADI1OUT1
G5	26	WADI1OUT2
G5	27	WADI1OUT3
G5	28	WADI1OUT4
G5	29	WADI1OUT5
G5	30	WADI1OUT6
G5	31	WADI1OUT7
G6	0	MCANA_FEVT0
G6	1	MCANA_FEVT1
G6	2	MCANA_FEVT2
G6	3	MCANB_FEVT0
G6	4	MCANB_FEVT1

Table 20-6. Output X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G6	5	MCANB_FEVT2
G6	6	MCANC_FEVT0
G6	7	MCANC_FEVT1
G6	8	MCANC_FEVT2
G6	9	MCAND_FEVT0
G6	10	MCAND_FEVT1
G6	11	MCAND_FEVT2
G6	12	MCANE_FEVT0
G6	13	MCANE_FEVT1
G6	14	MCANE_FEVT2
G6	15	MCANF_FEVT0
G6	16	MCANF_FEVT1
G6	17	MCANF_FEVT2
G6	18	Reserved
G6	19	Reserved
G6	20	Reserved
G6	21	Reserved
G6	22	Reserved
G6	23	Reserved
G6	24	WADI2OUT0
G6	25	WADI2OUT1
G6	26	WADI2OUT2
G6	27	WADI2OUT3
G6	28	WADI2OUT4
G6	29	WADI2OUT5
G6	30	WADI2OUT6
G6	31	WADI2OUT7
G7	0	CLB1_OUT0
G7	1	CLB1_OUT1
G7	2	CLB1_OUT2
G7	3	CLB1_OUT3
G7	4	CLB1_OUT4
G7	5	CLB1_OUT5
G7	6	CLB1_OUT6
G7	7	CLB1_OUT7
G7	8	CLB2_OUT0
G7	9	CLB2_OUT1
G7	10	CLB2_OUT2

Table 20-6. Output X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G7	11	CLB2_OUT3
G7	12	CLB2_OUT4
G7	13	CLB2_OUT5
G7	14	CLB2_OUT6
G7	15	CLB2_OUT7
G7	16	CLB3_OUT0
G7	17	CLB3_OUT1
G7	18	CLB3_OUT2
G7	19	CLB3_OUT3
G7	20	CLB3_OUT4
G7	21	CLB3_OUT5
G7	22	CLB3_OUT6
G7	23	CLB3_OUT7
G7	24	CLB4_OUT0
G7	25	CLB4_OUT1
G7	26	CLB4_OUT2
G7	27	CLB4_OUT3
G7	28	CLB4_OUT4
G7	29	CLB4_OUT5
G7	30	CLB4_OUT6
G7	31	CLB4_OUT7
G8	0	CLB5_OUT0
G8	1	CLB5_OUT1
G8	2	CLB5_OUT2
G8	3	CLB5_OUT3
G8	4	CLB5_OUT4
G8	5	CLB5_OUT5
G8	6	CLB5_OUT6
G8	7	CLB5_OUT7
G8	8	CLB6_OUT0
G8	9	CLB6_OUT1
G8	10	CLB6_OUT2
G8	11	CLB6_OUT3
G8	12	CLB6_OUT4
G8	13	CLB6_OUT5
G8	14	CLB6_OUT6
G8	15	CLB6_OUT7
G8	16	Reserved

Table 20-6. Output X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G8	17	Reserved
G8	18	Reserved
G8	19	Reserved
G8	20	Reserved
G8	21	Reserved
G8	22	Reserved
G8	23	Reserved
G8	24	Reserved
G8	25	Reserved
G8	26	Reserved
G8	27	Reserved
G8	28	Reserved
G8	29	Reserved
G8	30	Reserved
G8	31	Reserved
G9	0	CLBXBAR1
G9	1	CLBXBAR2
G9	2	CLBXBAR3
G9	3	CLBXBAR4
G9	4	CLBXBAR5
G9	5	CLBXBAR6
G9	6	CLBXBAR7
G9	7	CLBXBAR8
G9	8	ADCA_EXTMUXSEL0
G9	9	ADCA_EXTMUXSEL1
G9	10	ADCA_EXTMUXSEL2
G9	11	ADCA_EXTMUXSEL3
G9	12	ADCB_EXTMUXSEL0
G9	13	ADCB_EXTMUXSEL1
G9	14	ADCB_EXTMUXSEL2
G9	15	ADCB_EXTMUXSEL3
G9	16	ADCC_EXTMUXSEL0
G9	17	ADCC_EXTMUXSEL1
G9	18	ADCC_EXTMUXSEL2
G9	19	ADCC_EXTMUXSEL3
G9	20	ADCD_EXTMUXSEL0
G9	21	ADCD_EXTMUXSEL1
G9	22	ADCD_EXTMUXSEL2

Table 20-6. Output X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G9	23	ADCD_EXTMUXSEL3
G9	24	ADCE_EXTMUXSEL0
G9	25	ADCE_EXTMUXSEL1
G9	26	ADCE_EXTMUXSEL2
G9	27	ADCE_EXTMUXSEL3
G9	28	Reserved
G9	29	Reserved
G9	30	Reserved
G9	31	Reserved

20.3.4 X-BAR Flags

With the exception of the CMPSS signals, the ePWM X-BAR and the Output X-BAR have all of the same input signals. Due to the inputs being similar between the ePWM X-BAR, CLB X-BAR, and Output X-BAR, all X-BAR modules leverage a single set of input flags to indicate which input signals have been triggered. This allows software to check the input flags when an event occurs. See Figure 20-7 for more information. There is a bit allocated for each input signal in one of the XBARFLGx registers. The flag remains set until cleared through the appropriate XBARCLRx register.

Note

Not all input sources are routed to all X-BAR modules. Refer to the X-BAR specific configuration tables for exact connections.

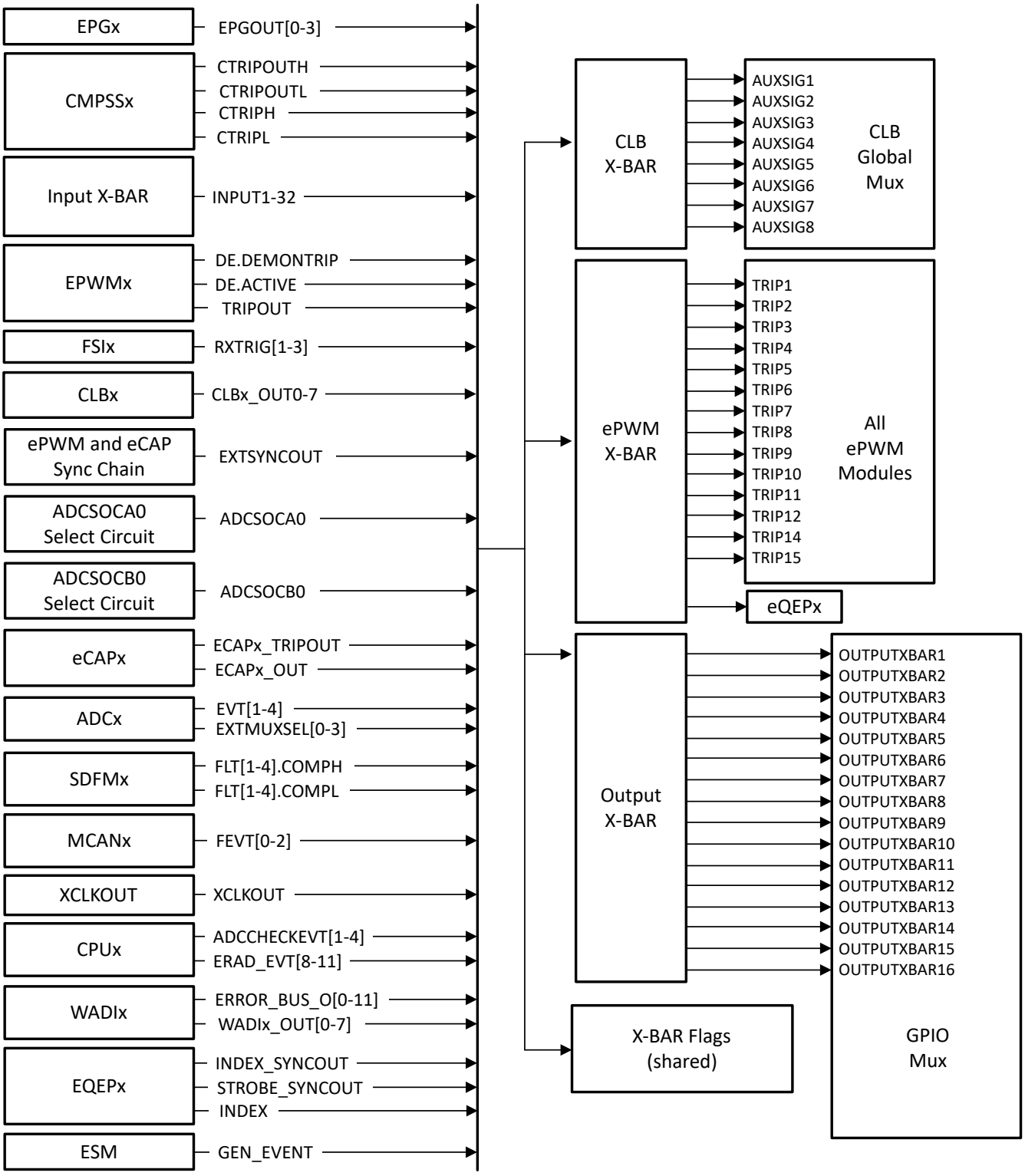


Figure 20-7. X-BAR Input Sources

20.4 Software

20.4.1 INPUT_XBAR Registers to Driverlib Functions

Table 20-7. INPUT_XBAR Registers to Driverlib Functions

File	Driverlib Function
INPUTSELECT(i)	
xbar.h	XBAR_setInputPin
xbar.h	XBAR_lockInput
INPUTSELECTLOCK1	
xbar.h	XBAR_lockInput
INPUTSELECTLOCK2	
xbar.h	XBAR_lockInput

20.4.2 EPWM_XBAR Registers to Driverlib Functions

Table 20-8. EPWM_XBAR Registers to Driverlib Functions

File	Driverlib Function
PWMXBAROUTINVERT	
-	
PWMXBARLOCK	
-	
PWMXBARG0SEL(i)	
xbar.c	XBAR_selectEpwmXbarInputSource
xbar.c	XBAR_deselectEpwmXbarInputSource
PWMXBARG1SEL(i)	
-	
PWMXBARG2SEL(i)	
-	
PWMXBARG3SEL(i)	
-	
PWMXBARG4SEL(i)	
-	
PWMXBARG5SEL(i)	
-	
PWMXBARG6SEL(i)	
-	
PWMXBARG7SEL(i)	
-	
PWMXBARG8SEL(i)	
-	
PWMXBARG9SEL(i)	
-	

20.4.3 CLB_XBAR Registers to Driverlib Functions

Table 20-9. CLB_XBAR Registers to Driverlib Functions

File	Driverlib Function
CLBXBAROUTINVERT	
-	

Table 20-9. CLB_XBAR Registers to Driverlib Functions (continued)

File	Driverlib Function
CLBXBARLOCK	
-	
CLBXBARG0SEL(i)	
xbar.c	XBAR_selectClbXbarInputSource
xbar.c	XBAR_deselectClbXbarInputSource
CLBXBARG1SEL(i)	
-	
CLBXBARG2SEL(i)	
-	
CLBXBARG3SEL(i)	
-	
CLBXBARG4SEL(i)	
-	
CLBXBARG5SEL(i)	
-	
CLBXBARG6SEL(i)	
-	
CLBXBARG7SEL(i)	
-	

20.4.4 OUTPUT_XBAR Registers to Driverlib Functions

Table 20-10. OUTPUT_XBAR Registers to Driverlib Functions

File	Driverlib Function
OUTPUTXBARSTATUS	
xbar.h	XBAR_getOutputSignalStatus
OUTPUTXBARFLAG	
xbar.h	XBAR_getOutputLatchStatus
xbar.h	XBAR_clearOutputLatch
xbar.h	XBAR_forceOutputLatch
xbar.h	XBAR_invertOutputSignalBeforeLatch
OUTPUTXBARFLAGCLEAR	
xbar.h	XBAR_clearOutputLatch
OUTPUTXBARFLAGFORCE	
xbar.h	XBAR_forceOutputLatch
OUTPUTXBARFLAGINVERT	
xbar.h	XBAR_invertOutputSignalBeforeLatch
OUTPUTXBAROUTLATCH	
xbar.h	XBAR_setOutputLatchMode
OUTPUTXBAROUTSTRETCH	
xbar.h	XBAR_selectOutputStretchSource
OUTPUTXBAROUTLENGTH	
xbar.h	XBAR_selectOutputStretchLength
OUTPUTXBAROUTINVERT	
-	
OUTPUTXBARLOCK	

Table 20-10. OUTPUT_XBAR Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
OUTPUTXBARG0SEL(i)	
xbar.c	XBAR_selectOutputXbarInputSource
xbar.c	XBAR_deselectOutputXbarInputSource
OUTPUTXBARG1SEL(i)	
-	
OUTPUTXBARG2SEL(i)	
-	
OUTPUTXBARG3SEL(i)	
-	
OUTPUTXBARG4SEL(i)	
-	
OUTPUTXBARG5SEL(i)	
-	
OUTPUTXBARG6SEL(i)	
-	
OUTPUTXBARG7SEL(i)	
-	
OUTPUTXBARG8SEL(i)	
-	
OUTPUTXBARG9SEL(i)	
-	

20.4.5 MDL_XBAR Registers to Driverlib Functions

Table 20-11. MDL_XBAR Registers to Driverlib Functions

File	Driverlib Function
MDLXBAROUTINVERT	
-	
MDLXBARLOCK	
-	
MDLXBARG0SEL(i)	
xbar.c	XBAR_selectMindbXbarInputSource
xbar.c	XBAR_deselectMindbXbarInputSource
MDLXBARG1SEL(i)	
-	
MDLXBARG2SEL(i)	
-	

20.4.6 ICL_XBAR Registers to Driverlib Functions

Table 20-12. ICL_XBAR Registers to Driverlib Functions

File	Driverlib Function
ICLXBAROUTINVERT	
-	
ICLXBARLOCK	
-	

Table 20-12. ICL_XBAR Registers to Driverlib Functions (continued)

File	Driverlib Function
ICLXBARG0SEL(i)	
xbar.c	XBAR_selectIclXbarInputSource
xbar.c	XBAR_deselectIclXbarInputSource
ICLXBARG1SEL(i)	
-	
ICLXBARG2SEL(i)	
-	

20.4.7 XBAR Registers to Driverlib Functions

Table 20-13. XBAR Registers to Driverlib Functions

File	Driverlib Function
FLG1	
xbar.c	XBAR_getInputFlagStatus
FLG2	
xbar.c	XBAR_getInputFlagStatus
FLG3	
xbar.c	XBAR_getInputFlagStatus
FLG4	
xbar.c	XBAR_getInputFlagStatus
FLG5	
xbar.c	XBAR_getInputFlagStatus
FLG6	
xbar.c	XBAR_getInputFlagStatus
FLG7	
xbar.c	XBAR_getInputFlagStatus
FLG8	
xbar.c	XBAR_getInputFlagStatus
FLG9	
xbar.c	XBAR_getInputFlagStatus
FLG10	
xbar.c	XBAR_getInputFlagStatus
FLG11	
xbar.c	XBAR_getInputFlagStatus
FLG12	
xbar.c	XBAR_getInputFlagStatus
FLG13	
xbar.c	XBAR_getInputFlagStatus
FLG14	
xbar.c	XBAR_getInputFlagStatus
FLG15	
xbar.c	XBAR_getInputFlagStatus
FLG16	
xbar.c	XBAR_getInputFlagStatus
FLG17	
xbar.c	XBAR_getInputFlagStatus

Table 20-13. XBAR Registers to Driverlib Functions (continued)

File	Driverlib Function
FLG18	
xbar.c	XBAR_getInputFlagStatus
CLR1	
xbar.c	XBAR_clearInputFlag
CLR2	
xbar.c	XBAR_clearInputFlag
CLR3	
xbar.c	XBAR_clearInputFlag
CLR4	
xbar.c	XBAR_clearInputFlag
CLR5	
xbar.c	XBAR_clearInputFlag
CLR6	
xbar.c	XBAR_clearInputFlag
CLR7	
xbar.c	XBAR_clearInputFlag
CLR8	
xbar.c	XBAR_clearInputFlag
CLR9	
xbar.c	XBAR_clearInputFlag
CLR10	
xbar.c	XBAR_clearInputFlag
CLR11	
xbar.c	XBAR_clearInputFlag
CLR12	
xbar.c	XBAR_clearInputFlag
CLR13	
xbar.c	XBAR_clearInputFlag
CLR14	
xbar.c	XBAR_clearInputFlag
CLR15	
xbar.c	XBAR_clearInputFlag
CLR16	
xbar.c	XBAR_clearInputFlag
CLR17	
xbar.c	XBAR_clearInputFlag
CLR18	
xbar.c	XBAR_clearInputFlag

20.4.8 XBAR Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/xbar

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

20.4.8.1 Input XBAR to Output XBAR Connection - SINGLE_CORE

FILE: xbar_ex1_input_output.c

Utilizes the Input XBAR and Output XBAR modules to connect one GPIO to another without the use of external jumpers.

External Connections

- GPIO0 is an input. Connect an external signal to this GPIO
- GPIO1 is an output. Probe this GPIO with an oscilloscope

The state of GPIO1 should always match the state of GPIO0 trigger) is routed to the input X-BAR, from which it is routed to TZ1.

20.4.8.2 Output XBAR Pulse Stretch - SINGLE_CORE

FILE: xbar_ex2_output_pulse_stretch.c

Reads an EPWM with a short duty cycle via an Input XBAR and connects it to the Output XBAR module. Enables pulse stretching in the Output XBAR to create a minimum duty cycle of 16 SYSCLKs.

External Connections

- GPIO0 and GPIO1 are EPWM outputs.
- GPIO2 is the pulse-stretched Output XBAR output.

20.5 XBAR Registers

This Section describes the XBAR Registers.

20.5.1 XBAR Base Address Table

Table 20-14. XBAR Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
INPUT_XBAR_REGS [1]	INPUTXBAR_BASE	0x301E_8000	-	YES	YES	YES	-	-	YES	YES
EPWM_XBAR_REGS [1]	EPWMXBAR_BASE	0x301E_9000	-	YES	YES	YES	-	-	YES	YES
CLB_XBAR_REGS [1]	CLBXBAR_BASE	0x301E_A000	-	YES	YES	YES	-	-	YES	YES
OUTPUTXBAR_REGS [1]	OUTPUTXBAR_BASE	0x301E_B000	-	YES	YES	YES	-	-	YES	YES
MDL_XBAR_REGS [1]	MDLXBAR_BASE	0x301E_C000	-	YES	YES	YES	-	-	YES	YES
ICL_XBAR_REGS [1]	ICLXBAR_BASE	0x301E_D000	-	YES	YES	YES	-	-	YES	YES
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR1_FLAGS_BASE	0x600C_0000	YES	YES	YES	YES	YES	YES	-	YES
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR2_FLAGS_BASE	0x600C_1000	YES	YES	YES	YES	YES	YES	-	YES
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR3_FLAGS_BASE	0x600C_2000	YES	YES	YES	YES	YES	YES	-	YES
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR4_FLAGS_BASE	0x600C_3000	YES	YES	YES	YES	YES	YES	-	YES
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR5_FLAGS_BASE	0x600C_4000	YES	YES	YES	YES	YES	YES	-	YES
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR6_FLAGS_BASE	0x600C_5000	YES	YES	YES	YES	YES	YES	-	YES

Table 20-14. XBAR Base Address Table (continued)

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR7_FLAGS_BASE	0x600C_6000	YES	YES	YES	YES	YES	YES	-	YES
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR8_FLAGS_BASE	0x600C_7000	YES	YES	YES	YES	YES	YES	-	YES
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR9_FLAGS_BASE	0x600C_8000	YES	YES	YES	YES	YES	YES	-	YES
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR10_FLAGS_BASE	0x600C_9000	YES	YES	YES	YES	YES	YES	-	YES
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR11_FLAGS_BASE	0x600C_A000	YES	YES	YES	YES	YES	YES	-	YES
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR12_FLAGS_BASE	0x600C_B000	YES	YES	YES	YES	YES	YES	-	YES
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR13_FLAGS_BASE	0x600C_C000	YES	YES	YES	YES	YES	YES	-	YES
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR14_FLAGS_BASE	0x600C_D000	YES	YES	YES	YES	YES	YES	-	YES
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR15_FLAGS_BASE	0x600C_E000	YES	YES	YES	YES	YES	YES	-	YES
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR16_FLAGS_BASE	0x600C_F000	YES	YES	YES	YES	YES	YES	-	YES
XBAR_REGS	XBAR_BASE	0x600E_0000	YES	YES	YES	YES	YES	YES	-	YES

(1) Registers writeable by CPU1.LINK0, CPU1.LINK1, CPU1.LINK2 only. All CPUs can read all registers in all LINKs. Debug write access only allowed if Zone0 or Zone1 are enabled for full debug by all CPUs. Debug reads always allowed. Register Read/Write access by HSM.

20.5.2 INPUT_XBAR_REGS Registers

Table 20-15 lists the memory-mapped registers for the INPUT_XBAR_REGS registers. All register offset addresses not listed in Table 20-15 should be considered as reserved locations and the register contents should not be modified.

Table 20-15. INPUT_XBAR_REGS Registers

Offset	Acronym	Register Name	Protection
0h + formula	INPUTSELECT_y	Input Select Register (GPIO0 to x)	
400h	INPUTSELECTLOCK1	Input Select Lock Register 1	
404h	INPUTSELECTLOCK2	Input Select Lock Register 2	

Complex bit access types are encoded to fit into small table cells. Table 20-16 shows the codes that are used for access types in this section.

Table 20-16. INPUT_XBAR_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

20.5.2.1 INPUTSELECT_y Register (Offset = 0h + formula) [Reset = 0000FFFEh]

INPUTSELECT_y is shown in Figure 20-8 and described in Table 20-17.

Return to the Summary Table.

Input Select Register (GPIO0 to x)

Offset = 0h + (y * 4h); where y = 0h to 3Fh

Figure 20-8. INPUTSELECT_y Register

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

Table 20-17. INPUTSELECT_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	SELECT	R/W	FFFEh	Select GPIO for INPUT{#} signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0xFFFD: '1' will be driven to the destination 0xFFFE: '1' will be driven to the destination 0xFFFF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0xFFFF) will cause the destination to be driven '1'. Reset type: XRSn

20.5.2.2 INPUTSELECTLOCK1 Register (Offset = 400h) [Reset = 00000000h]

INPUTSELECTLOCK1 is shown in Figure 20-9 and described in Table 20-18.

Return to the Summary Table.

Input Select Lock Register.
Any bit in this register, once set can only be cleared through SYSRSn. Write of 0 to any bit of this register has no effect. Reads to the registers which have LOCK protection are always allowed.

Figure 20-9. INPUTSELECTLOCK1 Register

31	30	29	28	27	26	25	24
INPUT32SELECT	INPUT31SELECT	INPUT30SELECT	INPUT29SELECT	INPUT28SELECT	INPUT27SELECT	INPUT26SELECT	INPUT25SELECT
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
INPUT24SELECT	INPUT23SELECT	INPUT22SELECT	INPUT21SELECT	INPUT20SELECT	INPUT19SELECT	INPUT18SELECT	INPUT17SELECT
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
INPUT16SELECT	INPUT15SELECT	INPUT14SELECT	INPUT13SELECT	INPUT12SELECT	INPUT11SELECT	INPUT10SELECT	INPUT9SELECT
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
INPUT8SELECT	INPUT7SELECT	INPUT6SELECT	INPUT5SELECT	INPUT4SELECT	INPUT3SELECT	INPUT2SELECT	INPUT1SELECT
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 20-18. INPUTSELECTLOCK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT32SELECT	R/WSonce	0h	Lock bit for INPUT32SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
30	INPUT31SELECT	R/WSonce	0h	Lock bit for INPUT31SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
29	INPUT30SELECT	R/WSonce	0h	Lock bit for INPUT30SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
28	INPUT29SELECT	R/WSonce	0h	Lock bit for INPUT29SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
27	INPUT28SELECT	R/WSonce	0h	Lock bit for INPUT28SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
26	INPUT27SELECT	R/WSonce	0h	Lock bit for INPUT27SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn

Table 20-18. INPUTSELECTLOCK1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	INPUT26SELECT	R/WSonce	0h	Lock bit for INPUT26SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
24	INPUT25SELECT	R/WSonce	0h	Lock bit for INPUT25SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
23	INPUT24SELECT	R/WSonce	0h	Lock bit for INPUT24SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
22	INPUT23SELECT	R/WSonce	0h	Lock bit for INPUT23SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
21	INPUT22SELECT	R/WSonce	0h	Lock bit for INPUT22SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
20	INPUT21SELECT	R/WSonce	0h	Lock bit for INPUT21SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
19	INPUT20SELECT	R/WSonce	0h	Lock bit for INPUT20SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
18	INPUT19SELECT	R/WSonce	0h	Lock bit for INPUT19SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
17	INPUT18SELECT	R/WSonce	0h	Lock bit for INPUT18SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
16	INPUT17SELECT	R/WSonce	0h	Lock bit for INPUT17SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
15	INPUT16SELECT	R/WSonce	0h	Lock bit for INPUT16SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
14	INPUT15SELECT	R/WSonce	0h	Lock bit for INPUT15SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
13	INPUT14SELECT	R/WSonce	0h	Lock bit for INPUT14SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
12	INPUT13SELECT	R/WSonce	0h	Lock bit for INPUT13SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn

Table 20-18. INPUTSELECTLOCK1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	INPUT12SELECT	R/WSonce	0h	Lock bit for INPUT12SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
10	INPUT11SELECT	R/WSonce	0h	Lock bit for INPUT11SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
9	INPUT10SELECT	R/WSonce	0h	Lock bit for INPUT10SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
8	INPUT9SELECT	R/WSonce	0h	Lock bit for INPUT9SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
7	INPUT8SELECT	R/WSonce	0h	Lock bit for INPUT8SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
6	INPUT7SELECT	R/WSonce	0h	Lock bit for INPUT7SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
5	INPUT6SELECT	R/WSonce	0h	Lock bit for INPUT6SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
4	INPUT5SELECT	R/WSonce	0h	Lock bit for INPUT5SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
3	INPUT4SELECT	R/WSonce	0h	Lock bit for INPUT4SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
2	INPUT3SELECT	R/WSonce	0h	Lock bit for INPUT3SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
1	INPUT2SELECT	R/WSonce	0h	Lock bit for INPUT2SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
0	INPUT1SELECT	R/WSonce	0h	Lock bit for INPUT1SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn

20.5.2.3 INPUTSELECTLOCK2 Register (Offset = 404h) [Reset = 00000000h]

INPUTSELECTLOCK2 is shown in Figure 20-10 and described in Table 20-19.

Return to the Summary Table.

Input Select Lock Register.
Any bit in this register, once set can only be cleared through SYSRSn. Write of 0 to any bit of this register has no effect. Reads to the registers which have LOCK protection are always allowed.

Figure 20-10. INPUTSELECTLOCK2 Register

31	30	29	28	27	26	25	24
INPUT64SELECT	INPUT63SELECT	INPUT62SELECT	INPUT61SELECT	INPUT60SELECT	INPUT59SELECT	INPUT58SELECT	INPUT57SELECT
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
INPUT56SELECT	INPUT55SELECT	INPUT54SELECT	INPUT53SELECT	INPUT52SELECT	INPUT51SELECT	INPUT50SELECT	INPUT49SELECT
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
INPUT48SELECT	INPUT47SELECT	INPUT46SELECT	INPUT45SELECT	INPUT44SELECT	INPUT43SELECT	INPUT42SELECT	INPUT41SELECT
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
INPUT40SELECT	INPUT39SELECT	INPUT38SELECT	INPUT37SELECT	INPUT36SELECT	INPUT35SELECT	INPUT34SELECT	INPUT33SELECT
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 20-19. INPUTSELECTLOCK2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT64SELECT	R/WSonce	0h	Lock bit for INPUT64SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
30	INPUT63SELECT	R/WSonce	0h	Lock bit for INPUT63SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
29	INPUT62SELECT	R/WSonce	0h	Lock bit for INPUT62SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
28	INPUT61SELECT	R/WSonce	0h	Lock bit for INPUT61SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
27	INPUT60SELECT	R/WSonce	0h	Lock bit for INPUT60SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
26	INPUT59SELECT	R/WSonce	0h	Lock bit for INPUT59SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn

Table 20-19. INPUTSELECTLOCK2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	INPUT58SELECT	R/WSonce	0h	Lock bit for INPUT58SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
24	INPUT57SELECT	R/WSonce	0h	Lock bit for INPUT57SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
23	INPUT56SELECT	R/WSonce	0h	Lock bit for INPUT56SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
22	INPUT55SELECT	R/WSonce	0h	Lock bit for INPUT55SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
21	INPUT54SELECT	R/WSonce	0h	Lock bit for INPUT54SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
20	INPUT53SELECT	R/WSonce	0h	Lock bit for INPUT53SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
19	INPUT52SELECT	R/WSonce	0h	Lock bit for INPUT52SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
18	INPUT51SELECT	R/WSonce	0h	Lock bit for INPUT51SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
17	INPUT50SELECT	R/WSonce	0h	Lock bit for INPUT50SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
16	INPUT49SELECT	R/WSonce	0h	Lock bit for INPUT49SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
15	INPUT48SELECT	R/WSonce	0h	Lock bit for INPUT48SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
14	INPUT47SELECT	R/WSonce	0h	Lock bit for INPUT47SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
13	INPUT46SELECT	R/WSonce	0h	Lock bit for INPUT46SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
12	INPUT45SELECT	R/WSonce	0h	Lock bit for INPUT45SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn

Table 20-19. INPUTSELECTLOCK2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	INPUT44SELECT	R/WSonce	0h	Lock bit for INPUT44SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
10	INPUT43SELECT	R/WSonce	0h	Lock bit for INPUT43SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
9	INPUT42SELECT	R/WSonce	0h	Lock bit for INPUT42SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
8	INPUT41SELECT	R/WSonce	0h	Lock bit for INPUT41SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
7	INPUT40SELECT	R/WSonce	0h	Lock bit for INPUT40SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
6	INPUT39SELECT	R/WSonce	0h	Lock bit for INPUT39SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
5	INPUT38SELECT	R/WSonce	0h	Lock bit for INPUT38SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
4	INPUT37SELECT	R/WSonce	0h	Lock bit for INPUT37SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
3	INPUT36SELECT	R/WSonce	0h	Lock bit for INPUT36SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
2	INPUT35SELECT	R/WSonce	0h	Lock bit for INPUT35SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
1	INPUT34SELECT	R/WSonce	0h	Lock bit for INPUT34SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
0	INPUT33SELECT	R/WSonce	0h	Lock bit for INPUT33SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn

20.5.3 EPWM_XBAR_REGS Registers

Table 20-20 lists the memory-mapped registers for the EPWM_XBAR_REGS registers. All register offset addresses not listed in Table 20-20 should be considered as reserved locations and the register contents should not be modified.

Table 20-20. EPWM_XBAR_REGS Registers

Offset	Acronym	Register Name	Protection
30h	PWMXBAROutInvert	Output Signal Invert Select	
80h	PWMXBARLock	Configuration Lock register	
100h + formula	PWMXBARG0SEL_j	PWMXBAR G0 Input Select	
104h + formula	PWMXBARG1SEL_j	PWMXBAR G1 Input Select	
108h + formula	PWMXBARG2SEL_j	PWMXBAR G2 Input Select	
10Ch + formula	PWMXBARG3SEL_j	PWMXBAR G3 Input Select	
110h + formula	PWMXBARG4SEL_j	PWMXBAR G4 Input Select	
114h + formula	PWMXBARG5SEL_j	PWMXBAR G5 Input Select	
118h + formula	PWMXBARG6SEL_j	PWMXBAR G6 Input Select	
11Ch + formula	PWMXBARG7SEL_j	PWMXBAR G7 Input Select	
120h + formula	PWMXBARG8SEL_j	PWMXBAR G8 Input Select	
124h + formula	PWMXBARG9SEL_j	PWMXBAR G9 Input Select	

Complex bit access types are encoded to fit into small table cells. Table 20-21 shows the codes that are used for access types in this section.

Table 20-21. EPWM_XBAR_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.

Table 20-21. EPWM_XBAR_REGS Access Type Codes (continued)

Access Type	Code	Description
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

20.5.3.1 PWMXBAROutInvert Register (Offset = 30h) [Reset = 00000000h]

PWMXBAROutInvert is shown in Figure 20-11 and described in Table 20-22.

Return to the Summary Table.

Output Signal Invert Select

Figure 20-11. PWMXBAROutInvert Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
XBAR16	XBAR15	XBAR14	XBAR13	XBAR12	XBAR11	XBAR10	XBAR9
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
XBAR8	XBAR7	XBAR6	XBAR5	XBAR4	XBAR3	XBAR2	XBAR1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-22. PWMXBAROutInvert Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	XBAR16	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
14	XBAR15	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
13	XBAR14	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
12	XBAR13	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
11	XBAR12	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
10	XBAR11	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
9	XBAR10	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn

Table 20-22. PWMXBAROutInvert Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	XBAR9	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
7	XBAR8	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
6	XBAR7	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
5	XBAR6	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
4	XBAR5	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
3	XBAR4	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
2	XBAR3	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
1	XBAR2	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
0	XBAR1	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn

20.5.3.2 PWMXBARLock Register (Offset = 80h) [Reset = 00000000h]

PWMXBARLock is shown in Figure 20-12 and described in Table 20-23.

Return to the Summary Table.

Configuration Lock register

Figure 20-12. PWMXBARLock Register

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

Table 20-23. PWMXBARLock Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Bit-0 of this register can be set only if KEY= 0x5a5a Reset type: XRSn
15-1	RESERVED	R-0	0h	Reserved
0	LOCK	R/WSonce	0h	Locks the configuration for EPWM-XBAR. Once the configuration is locked, writes to the below registers for EPWM-XBAR is blocked. Registers Affected by the LOCK mechanism: PWMXBAR Gx SEL PWMXBAR.Invert 0: Writes to the above registers are allowed 1: Writes to the above registers are blocked Note: [1] LOCK mechanism only apples to writes. Reads are never blocked. Reset type: XRSn

20.5.3.3 PWMXBARG0SEL_j Register (Offset = 100h + formula) [Reset = 00000000h]

PWMXBARG0SEL_j is shown in Figure 20-13 and described in Table 20-24.

Return to the Summary Table.

PWMXBAR G0 Input Select

Offset = 100h + (j * 40h); where j = 0h to Fh

Figure 20-13. PWMXBARG0SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-24. PWMXBARG0SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-24. PWMXBARG0SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-24. PWMXBARG0SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.3.4 PWMXBARG1SEL_j Register (Offset = 104h + formula) [Reset = 00000000h]

PWMXBARG1SEL_j is shown in Figure 20-14 and described in Table 20-25.

Return to the Summary Table.

PWMXBAR G1 Input Select

Offset = 104h + (j * 40h); where j = 0h to Fh

Figure 20-14. PWMXBARG1SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-25. PWMXBARG1SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-25. PWMXBARG1SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-25. PWMXBARG1SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.3.5 PWMXBARG2SEL_j Register (Offset = 108h + formula) [Reset = 00000000h]

PWMXBARG2SEL_j is shown in Figure 20-15 and described in Table 20-26.

Return to the Summary Table.

PWMXBAR G2 Input Select

Offset = 108h + (j * 40h); where j = 0h to Fh

Figure 20-15. PWMXBARG2SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-26. PWMXBARG2SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-26. PWMXBARG2SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-26. PWMXBARG2SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.3.6 PWMXBARG3SEL_j Register (Offset = 10Ch + formula) [Reset = 00000000h]

PWMXBARG3SEL_j is shown in Figure 20-16 and described in Table 20-27.

Return to the Summary Table.

PWMXBAR G3 Input Select

Offset = 10Ch + (j * 40h); where j = 0h to Fh

Figure 20-16. PWMXBARG3SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-27. PWMXBARG3SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-27. PWMXBARG3SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-27. PWMXBARG3SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.3.7 PWMXBARG4SEL_j Register (Offset = 110h + formula) [Reset = 00000000h]

PWMXBARG4SEL_j is shown in Figure 20-17 and described in Table 20-28.

Return to the Summary Table.

PWMXBAR G4 Input Select

Offset = 110h + (j * 40h); where j = 0h to Fh

Figure 20-17. PWMXBARG4SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-28. PWMXBARG4SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-28. PWMXBARG4SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-28. PWMXBARG4SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.3.8 PWMXBARG5SEL_j Register (Offset = 114h + formula) [Reset = 00000000h]

PWMXBARG5SEL_j is shown in Figure 20-18 and described in Table 20-29.

Return to the Summary Table.

PWMXBAR G5 Input Select

Offset = 114h + (j * 40h); where j = 0h to Fh

Figure 20-18. PWMXBARG5SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-29. PWMXBARG5SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-29. PWMXBARG5SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-29. PWMXBARG5SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.3.9 PWMXBARG6SEL_j Register (Offset = 118h + formula) [Reset = 00000000h]

PWMXBARG6SEL_j is shown in Figure 20-19 and described in Table 20-30.

Return to the Summary Table.

PWMXBAR G6 Input Select

Offset = 118h + (j * 40h); where j = 0h to Fh

Figure 20-19. PWMXBARG6SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-30. PWMXBARG6SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-30. PWMXBARG6SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-30. PWMXBARG6SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.3.10 PWMXBARG7SEL_j Register (Offset = 11Ch + formula) [Reset = 00000000h]

PWMXBARG7SEL_j is shown in Figure 20-20 and described in Table 20-31.

Return to the Summary Table.

PWMXBAR G7 Input Select

Offset = 11Ch + (j * 40h); where j = 0h to Fh

Figure 20-20. PWMXBARG7SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-31. PWMXBARG7SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-31. PWMXBARG7SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-31. PWMXBARG7SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.3.11 PWMXBARG8SEL_j Register (Offset = 120h + formula) [Reset = 00000000h]

PWMXBARG8SEL_j is shown in Figure 20-21 and described in Table 20-32.

Return to the Summary Table.

PWMXBAR G8 Input Select

Offset = 120h + (j * 40h); where j = 0h to Fh

Figure 20-21. PWMXBARG8SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-32. PWMXBARG8SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-32. PWMXBARG8SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-32. PWMXBARG8SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.3.12 PWMXBARG9SEL_j Register (Offset = 124h + formula) [Reset = 00000000h]

PWMXBARG9SEL_j is shown in Figure 20-22 and described in Table 20-33.

Return to the Summary Table.

PWMXBAR G9 Input Select

Offset = 124h + (j * 40h); where j = 0h to Fh

Figure 20-22. PWMXBARG9SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-33. PWMXBARG9SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-33. PWMXBARG9SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-33. PWMXBARG9SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.4 CLB_XBAR_REGS Registers

Table 20-34 lists the memory-mapped registers for the CLB_XBAR_REGS registers. All register offset addresses not listed in Table 20-34 should be considered as reserved locations and the register contents should not be modified.

Table 20-34. CLB_XBAR_REGS Registers

Offset	Acronym	Register Name	Protection
30h	CLBXBAROutInvert	Output Signal Invert Select	
80h	CLBXBARLock	Configuration Lock register	
100h + formula	CLBXBARG0SEL_j	CLBXBAR G0 Input Select	
104h + formula	CLBXBARG1SEL_j	CLBXBAR G1 Input Select	
108h + formula	CLBXBARG2SEL_j	CLBXBAR G2 Input Select	
10Ch + formula	CLBXBARG3SEL_j	CLBXBAR G3 Input Select	
110h + formula	CLBXBARG4SEL_j	CLBXBAR G4 Input Select	
114h + formula	CLBXBARG5SEL_j	CLBXBAR G5 Input Select	
118h + formula	CLBXBARG6SEL_j	CLBXBAR G6 Input Select	
11Ch + formula	CLBXBARG7SEL_j	CLBXBAR G7 Input Select	

Complex bit access types are encoded to fit into small table cells. Table 20-35 shows the codes that are used for access types in this section.

Table 20-35. CLB_XBAR_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

20.5.4.1 CLBXBAROutInvert Register (Offset = 30h) [Reset = 00000000h]

CLBXBAROutInvert is shown in Figure 20-23 and described in Table 20-36.

Return to the Summary Table.

Output Signal Invert Select

Figure 20-23. CLBXBAROutInvert Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
XBAR8	XBAR7	XBAR6	XBAR5	XBAR4	XBAR3	XBAR2	XBAR1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-36. CLBXBAROutInvert Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	XBAR8	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
6	XBAR7	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
5	XBAR6	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
4	XBAR5	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
3	XBAR4	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
2	XBAR3	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
1	XBAR2	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn

Table 20-36. CLBXBAROutInvert Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	XBAR1	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn

20.5.4.2 CLBXBARLock Register (Offset = 80h) [Reset = 00000000h]

CLBXBARLock is shown in Figure 20-24 and described in Table 20-37.

Return to the Summary Table.

Configuration Lock register

Figure 20-24. CLBXBARLock Register

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

Table 20-37. CLBXBARLock Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Bit-0 of this register can be set only if KEY= 0x5a5a Reset type: XRSn
15-1	RESERVED	R-0	0h	Reserved
0	LOCK	R/WSonce	0h	Locks the configuration for ECLB-XBAR. Once the configuration is locked, writes to the below registers for ECLB-XBAR is blocked. Registers Affected by the LOCK mechanism: CLBXBAR Gx SEL CLBXBAR.Invert 0: Writes to the above registers are allowed 1: Writes to the above registers are blocked Note: [1] LOCK mechanism only apples to writes. Reads are never blocked. Reset type: XRSn

20.5.4.3 CLBXBARG0SEL_j Register (Offset = 100h + formula) [Reset = 00000000h]

CLBXBARG0SEL_j is shown in Figure 20-25 and described in Table 20-38.

Return to the Summary Table.

CLBXBAR G0 Input Select

Offset = 100h + (j * 40h); where j = 0h to 7h

Figure 20-25. CLBXBARG0SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-38. CLBXBARG0SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-38. CLBXBARG0SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-38. CLBXBARG0SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.4.4 CLBXBARG1SEL_j Register (Offset = 104h + formula) [Reset = 00000000h]

CLBXBARG1SEL_j is shown in Figure 20-26 and described in Table 20-39.

Return to the Summary Table.

CLBXBAR G1 Input Select

Offset = 104h + (j * 40h); where j = 0h to 7h

Figure 20-26. CLBXBARG1SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-39. CLBXBARG1SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-39. CLBXBARG1SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-39. CLBXBARG1SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.4.5 CLBXBARG2SEL_j Register (Offset = 108h + formula) [Reset = 00000000h]

CLBXBARG2SEL_j is shown in Figure 20-27 and described in Table 20-40.

Return to the Summary Table.

CLBXBAR G2 Input Select

Offset = 108h + (j * 40h); where j = 0h to 7h

Figure 20-27. CLBXBARG2SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-40. CLBXBARG2SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-40. CLBXBARG2SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-40. CLBXBARG2SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.4.6 CLBXBARG3SEL_j Register (Offset = 10Ch + formula) [Reset = 00000000h]

CLBXBARG3SEL_j is shown in Figure 20-28 and described in Table 20-41.

Return to the Summary Table.

CLBXBAR G3 Input Select

Offset = 10Ch + (j * 40h); where j = 0h to 7h

Figure 20-28. CLBXBARG3SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-41. CLBXBARG3SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-41. CLBXBARG3SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-41. CLBXBARG3SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.4.7 CLBXBARG4SEL_j Register (Offset = 110h + formula) [Reset = 00000000h]

CLBXBARG4SEL_j is shown in Figure 20-29 and described in Table 20-42.

Return to the Summary Table.

CLBXBAR G4 Input Select

Offset = 110h + (j * 40h); where j = 0h to 7h

Figure 20-29. CLBXBARG4SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-42. CLBXBARG4SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-42. CLBXBARG4SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-42. CLBXBARG4SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.4.8 CLBXBARG5SEL_j Register (Offset = 114h + formula) [Reset = 00000000h]

CLBXBARG5SEL_j is shown in Figure 20-30 and described in Table 20-43.

Return to the Summary Table.

CLBXBAR G5 Input Select

Offset = 114h + (j * 40h); where j = 0h to 7h

Figure 20-30. CLBXBARG5SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-43. CLBXBARG5SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-43. CLBXBARG5SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-43. CLBXBARG5SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.4.9 CLBXBARG6SEL_j Register (Offset = 118h + formula) [Reset = 00000000h]

CLBXBARG6SEL_j is shown in Figure 20-31 and described in Table 20-44.

Return to the Summary Table.

CLBXBAR G6 Input Select

Offset = 118h + (j * 40h); where j = 0h to 7h

Figure 20-31. CLBXBARG6SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-44. CLBXBARG6SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-44. CLBXBARG6SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-44. CLBXBARG6SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.4.10 CLBXBARG7SEL_j Register (Offset = 11Ch + formula) [Reset = 00000000h]

CLBXBARG7SEL_j is shown in Figure 20-32 and described in Table 20-45.

Return to the Summary Table.

CLBXBAR G7 Input Select

Offset = 11Ch + (j * 40h); where j = 0h to 7h

Figure 20-32. CLBXBARG7SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-45. CLBXBARG7SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-45. CLBXBARG7SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-45. CLBXBARG7SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.5 OUTPUTXBAR_REGS Registers

Table 20-46 lists the memory-mapped registers for the OUTPUTXBAR_REGS registers. All register offset addresses not listed in Table 20-46 should be considered as reserved locations and the register contents should not be modified.

Table 20-46. OUTPUTXBAR_REGS Registers

Offset	Acronym	Register Name	Protection
14h	OUTPUTXBARFlagInvert	Output Signal inversion register	
24h	OUTPUTXBAROutLatch	Output Signal Select Latch	
28h	OUTPUTXBAROutStretch	Output Signal Stretched Pulse Version Select	
2Ch	OUTPUTXBAROutLength	Output Signal Stretched Pulse Length Select	
30h	OUTPUTXBAROutInvert	Output Signal Invert Select	
80h	OUTPUTXBARLock	Configuration Lock register	
100h + formula	OUTPUTXBARG0SEL_j	OUTPUTXBAR G0 Input Select	
104h + formula	OUTPUTXBARG1SEL_j	OUTPUTXBAR G1 Input Select	
108h + formula	OUTPUTXBARG2SEL_j	OUTPUTXBAR G2 Input Select	
10Ch + formula	OUTPUTXBARG3SEL_j	OUTPUTXBAR G3 Input Select	
110h + formula	OUTPUTXBARG4SEL_j	OUTPUTXBAR G4 Input Select	
114h + formula	OUTPUTXBARG5SEL_j	OUTPUTXBAR G5 Input Select	
118h + formula	OUTPUTXBARG6SEL_j	OUTPUTXBAR G6 Input Select	
11Ch + formula	OUTPUTXBARG7SEL_j	OUTPUTXBAR G7 Input Select	
120h + formula	OUTPUTXBARG8SEL_j	OUTPUTXBAR G8 Input Select	
124h + formula	OUTPUTXBARG9SEL_j	OUTPUTXBAR G9 Input Select	

Complex bit access types are encoded to fit into small table cells. Table 20-47 shows the codes that are used for access types in this section.

Table 20-47. OUTPUTXBAR_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		

Table 20-47. OUTPUTXBAR_REGS Access Type Codes (continued)

Access Type	Code	Description
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

20.5.5.1 OUTPUTXBARFlagInvert Register (Offset = 14h) [Reset = 00000000h]

OUTPUTXBARFlagInvert is shown in Figure 20-33 and described in Table 20-48.

Return to the Summary Table.

Output Signal inversion register

Figure 20-33. OUTPUTXBARFlagInvert Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
XBAR16	XBAR15	XBAR14	XBAR13	XBAR12	XBAR11	XBAR10	XBAR9
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
XBAR8	XBAR7	XBAR6	XBAR5	XBAR4	XBAR3	XBAR2	XBAR1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-48. OUTPUTXBARFlagInvert Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	XBAR16	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
14	XBAR15	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
13	XBAR14	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
12	XBAR13	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
11	XBAR12	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
10	XBAR11	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
9	XBAR10	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn

Table 20-48. OUTPUTXBARFlagInvert Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	XBAR9	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
7	XBAR8	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
6	XBAR7	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
5	XBAR6	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
4	XBAR5	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
3	XBAR4	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
2	XBAR3	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
1	XBAR2	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
0	XBAR1	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn

20.5.5.2 OUTPUTXBAROutLatch Register (Offset = 24h) [Reset = 00000000h]

OUTPUTXBAROutLatch is shown in Figure 20-34 and described in Table 20-49.

Return to the Summary Table.

Output Signal Select Latch

Figure 20-34. OUTPUTXBAROutLatch Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
XBAR16	XBAR15	XBAR14	XBAR13	XBAR12	XBAR11	XBAR10	XBAR9
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
XBAR8	XBAR7	XBAR6	XBAR5	XBAR4	XBAR3	XBAR2	XBAR1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-49. OUTPUTXBAROutLatch Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	XBAR16	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
14	XBAR15	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
13	XBAR14	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
12	XBAR13	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
11	XBAR12	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
10	XBAR11	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
9	XBAR10	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn

Table 20-49. OUTPUTXBAROutLatch Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	XBAR9	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
7	XBAR8	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
6	XBAR7	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
5	XBAR6	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
4	XBAR5	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
3	XBAR4	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
2	XBAR3	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
1	XBAR2	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
0	XBAR1	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn

20.5.5.3 OUTPUTXBAROutStretch Register (Offset = 28h) [Reset = 00000000h]

OUTPUTXBAROutStretch is shown in Figure 20-35 and described in Table 20-50.

Return to the Summary Table.

Output Signal Stretched Pulse Version Select

Figure 20-35. OUTPUTXBAROutStretch Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
XBAR16	XBAR15	XBAR14	XBAR13	XBAR12	XBAR11	XBAR10	XBAR9
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
XBAR8	XBAR7	XBAR6	XBAR5	XBAR4	XBAR3	XBAR2	XBAR1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-50. OUTPUTXBAROutStretch Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	XBAR16	R/W	0h	Output Signal Stretched Pulse Version Select: 0 Non Stretched Signal Selected As Output 1 Stretched Pulse Selected As Output Reset type: XRSn
14	XBAR15	R/W	0h	Output Signal Stretched Pulse Version Select: 0 Non Stretched Signal Selected As Output 1 Stretched Pulse Selected As Output Reset type: XRSn
13	XBAR14	R/W	0h	Output Signal Stretched Pulse Version Select: 0 Non Stretched Signal Selected As Output 1 Stretched Pulse Selected As Output Reset type: XRSn
12	XBAR13	R/W	0h	Output Signal Stretched Pulse Version Select: 0 Non Stretched Signal Selected As Output 1 Stretched Pulse Selected As Output Reset type: XRSn
11	XBAR12	R/W	0h	Output Signal Stretched Pulse Version Select: 0 Non Stretched Signal Selected As Output 1 Stretched Pulse Selected As Output Reset type: XRSn
10	XBAR11	R/W	0h	Output Signal Stretched Pulse Version Select: 0 Non Stretched Signal Selected As Output 1 Stretched Pulse Selected As Output Reset type: XRSn
9	XBAR10	R/W	0h	Output Signal Stretched Pulse Version Select: 0 Non Stretched Signal Selected As Output 1 Stretched Pulse Selected As Output Reset type: XRSn

Table 20-50. OUTPUTXBAROutStretch Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	XBAR9	R/W	0h	Output Signal Stretched Pulse Version Select: 0 Non Stretched Signal Selected As Output 1 Stretched Pulse Selected As Output Reset type: XRSn
7	XBAR8	R/W	0h	Output Signal Stretched Pulse Version Select: 0 Non Stretched Signal Selected As Output 1 Stretched Pulse Selected As Output Reset type: XRSn
6	XBAR7	R/W	0h	Output Signal Stretched Pulse Version Select: 0 Non Stretched Signal Selected As Output 1 Stretched Pulse Selected As Output Reset type: XRSn
5	XBAR6	R/W	0h	Output Signal Stretched Pulse Version Select: 0 Non Stretched Signal Selected As Output 1 Stretched Pulse Selected As Output Reset type: XRSn
4	XBAR5	R/W	0h	Output Signal Stretched Pulse Version Select: 0 Non Stretched Signal Selected As Output 1 Stretched Pulse Selected As Output Reset type: XRSn
3	XBAR4	R/W	0h	Output Signal Stretched Pulse Version Select: 0 Non Stretched Signal Selected As Output 1 Stretched Pulse Selected As Output Reset type: XRSn
2	XBAR3	R/W	0h	Output Signal Stretched Pulse Version Select: 0 Non Stretched Signal Selected As Output 1 Stretched Pulse Selected As Output Reset type: XRSn
1	XBAR2	R/W	0h	Output Signal Stretched Pulse Version Select: 0 Non Stretched Signal Selected As Output 1 Stretched Pulse Selected As Output Reset type: XRSn
0	XBAR1	R/W	0h	Output Signal Stretched Pulse Version Select: 0 Non Stretched Signal Selected As Output 1 Stretched Pulse Selected As Output Reset type: XRSn

20.5.5.4 OUTPUTXBAROutLength Register (Offset = 2Ch) [Reset = 00000000h]

OUTPUTXBAROutLength is shown in Figure 20-36 and described in Table 20-51.

Return to the Summary Table.

Output Signal Stretched Pulse Length Select

Figure 20-36. OUTPUTXBAROutLength Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
XBAR16	XBAR15	XBAR14	XBAR13	XBAR12	XBAR11	XBAR10	XBAR9
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
XBAR8	XBAR7	XBAR6	XBAR5	XBAR4	XBAR3	XBAR2	XBAR1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-51. OUTPUTXBAROutLength Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	XBAR16	R/W	0h	Output Signal Stretched Pulse Length Select: 0 16 SYSCLK Stretch 1 32 SYSCLK Stretch Reset type: XRSn
14	XBAR15	R/W	0h	Output Signal Stretched Pulse Length Select: 0 16 SYSCLK Stretch 1 32 SYSCLK Stretch Reset type: XRSn
13	XBAR14	R/W	0h	Output Signal Stretched Pulse Length Select: 0 16 SYSCLK Stretch 1 32 SYSCLK Stretch Reset type: XRSn
12	XBAR13	R/W	0h	Output Signal Stretched Pulse Length Select: 0 16 SYSCLK Stretch 1 32 SYSCLK Stretch Reset type: XRSn
11	XBAR12	R/W	0h	Output Signal Stretched Pulse Length Select: 0 16 SYSCLK Stretch 1 32 SYSCLK Stretch Reset type: XRSn
10	XBAR11	R/W	0h	Output Signal Stretched Pulse Length Select: 0 16 SYSCLK Stretch 1 32 SYSCLK Stretch Reset type: XRSn
9	XBAR10	R/W	0h	Output Signal Stretched Pulse Length Select: 0 16 SYSCLK Stretch 1 32 SYSCLK Stretch Reset type: XRSn

Table 20-51. OUTPUTXBAROutLength Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	XBAR9	R/W	0h	Output Signal Stretched Pulse Length Select: 0 16 SYSCLK Stretch 1 32 SYSCLK Stretch Reset type: XRSn
7	XBAR8	R/W	0h	Output Signal Stretched Pulse Length Select: 0 16 SYSCLK Stretch 1 32 SYSCLK Stretch Reset type: XRSn
6	XBAR7	R/W	0h	Output Signal Stretched Pulse Length Select: 0 16 SYSCLK Stretch 1 32 SYSCLK Stretch Reset type: XRSn
5	XBAR6	R/W	0h	Output Signal Stretched Pulse Length Select: 0 16 SYSCLK Stretch 1 32 SYSCLK Stretch Reset type: XRSn
4	XBAR5	R/W	0h	Output Signal Stretched Pulse Length Select: 0 16 SYSCLK Stretch 1 32 SYSCLK Stretch Reset type: XRSn
3	XBAR4	R/W	0h	Output Signal Stretched Pulse Length Select: 0 16 SYSCLK Stretch 1 32 SYSCLK Stretch Reset type: XRSn
2	XBAR3	R/W	0h	Output Signal Stretched Pulse Length Select: 0 16 SYSCLK Stretch 1 32 SYSCLK Stretch Reset type: XRSn
1	XBAR2	R/W	0h	Output Signal Stretched Pulse Length Select: 0 16 SYSCLK Stretch 1 32 SYSCLK Stretch Reset type: XRSn
0	XBAR1	R/W	0h	Output Signal Stretched Pulse Length Select: 0 16 SYSCLK Stretch 1 32 SYSCLK Stretch Reset type: XRSn

20.5.5.5 OUTPUTXBAROutInvert Register (Offset = 30h) [Reset = 00000000h]

OUTPUTXBAROutInvert is shown in Figure 20-37 and described in Table 20-52.

Return to the Summary Table.

Output Signal Invert Select

Figure 20-37. OUTPUTXBAROutInvert Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
XBAR16	XBAR15	XBAR14	XBAR13	XBAR12	XBAR11	XBAR10	XBAR9
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
XBAR8	XBAR7	XBAR6	XBAR5	XBAR4	XBAR3	XBAR2	XBAR1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-52. OUTPUTXBAROutInvert Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	XBAR16	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
14	XBAR15	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
13	XBAR14	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
12	XBAR13	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
11	XBAR12	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
10	XBAR11	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
9	XBAR10	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn

Table 20-52. OUTPUTXBAROutInvert Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	XBAR9	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
7	XBAR8	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
6	XBAR7	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
5	XBAR6	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
4	XBAR5	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
3	XBAR4	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
2	XBAR3	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
1	XBAR2	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
0	XBAR1	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn

20.5.5.6 OUTPUTXBARLock Register (Offset = 80h) [Reset = 00000000h]

OUTPUTXBARLock is shown in Figure 20-38 and described in Table 20-53.

Return to the Summary Table.

Configuration Lock register

Figure 20-38. OUTPUTXBARLock Register

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

Table 20-53. OUTPUTXBARLock Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Bit-0 of this register can be set only if KEY= 0x5a5a Reset type: XRSn
15-1	RESERVED	R-0	0h	Reserved
0	LOCK	R/WSonce	0h	Locks the configuration for EOUTPUT-XBAR. Once the configuration is locked, writes to the below registers for EOUTPUT-XBAR is blocked. Registers Affected by the LOCK mechanism: OUTPUTXBAR Gx SEL OUTPUTXBAR.Invert 0: Writes to the above registers are allowed 1: Writes to the above registers are blocked Note: [1] LOCK mechanism only apples to writes. Reads are never blocked. Reset type: XRSn

20.5.5.7 OUTPUTXBARG0SEL_j Register (Offset = 100h + formula) [Reset = 00000000h]

OUTPUTXBARG0SEL_j is shown in Figure 20-39 and described in Table 20-54.

Return to the Summary Table.

OUTPUTXBAR G0 Input Select

Offset = 100h + (j * 40h); where j = 0h to Fh

Figure 20-39. OUTPUTXBARG0SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-54. OUTPUTXBARG0SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-54. OUTPUTXBARG0SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-54. OUTPUTXBARG0SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.5.8 OUTPUTXBARG1SEL_j Register (Offset = 104h + formula) [Reset = 00000000h]

OUTPUTXBARG1SEL_j is shown in Figure 20-40 and described in Table 20-55.

Return to the Summary Table.

OUTPUTXBAR G1 Input Select

Offset = 104h + (j * 40h); where j = 0h to Fh

Figure 20-40. OUTPUTXBARG1SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-55. OUTPUTXBARG1SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-55. OUTPUTXBARG1SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-55. OUTPUTXBARG1SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.5.9 OUTPUTXBARG2SEL_j Register (Offset = 108h + formula) [Reset = 00000000h]

OUTPUTXBARG2SEL_j is shown in Figure 20-41 and described in Table 20-56.

Return to the Summary Table.

OUTPUTXBAR G2 Input Select

Offset = 108h + (j * 40h); where j = 0h to Fh

Figure 20-41. OUTPUTXBARG2SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-56. OUTPUTXBARG2SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-56. OUTPUTXBARG2SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-56. OUTPUTXBARG2SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.5.10 OUTPUTXBARG3SEL_j Register (Offset = 10Ch + formula) [Reset = 00000000h]

OUTPUTXBARG3SEL_j is shown in Figure 20-42 and described in Table 20-57.

Return to the Summary Table.

OUTPUTXBAR G3 Input Select

Offset = 10Ch + (j * 40h); where j = 0h to Fh

Figure 20-42. OUTPUTXBARG3SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-57. OUTPUTXBARG3SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-57. OUTPUTXBARG3SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-57. OUTPUTXBARG3SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.5.11 OUTPUTXBARG4SEL_j Register (Offset = 110h + formula) [Reset = 00000000h]

OUTPUTXBARG4SEL_j is shown in Figure 20-43 and described in Table 20-58.

Return to the Summary Table.

OUTPUTXBAR G4 Input Select

Offset = 110h + (j * 40h); where j = 0h to Fh

Figure 20-43. OUTPUTXBARG4SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-58. OUTPUTXBARG4SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-58. OUTPUTXBARG4SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-58. OUTPUTXBARG4SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.5.12 OUTPUTXBARG5SEL_j Register (Offset = 114h + formula) [Reset = 00000000h]

OUTPUTXBARG5SEL_j is shown in Figure 20-44 and described in Table 20-59.

Return to the Summary Table.

OUTPUTXBAR G5 Input Select

Offset = 114h + (j * 40h); where j = 0h to Fh

Figure 20-44. OUTPUTXBARG5SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-59. OUTPUTXBARG5SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-59. OUTPUTXBARG5SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-59. OUTPUTXBARG5SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.5.13 OUTPUTXBARG6SEL_j Register (Offset = 118h + formula) [Reset = 00000000h]

OUTPUTXBARG6SEL_j is shown in Figure 20-45 and described in Table 20-60.

Return to the Summary Table.

OUTPUTXBAR G6 Input Select

Offset = 118h + (j * 40h); where j = 0h to Fh

Figure 20-45. OUTPUTXBARG6SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-60. OUTPUTXBARG6SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-60. OUTPUTXBARG6SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-60. OUTPUTXBARG6SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.5.14 OUTPUTXBARG7SEL_j Register (Offset = 11Ch + formula) [Reset = 00000000h]

OUTPUTXBARG7SEL_j is shown in Figure 20-46 and described in Table 20-61.

Return to the Summary Table.

OUTPUTXBAR G7 Input Select

Offset = 11Ch + (j * 40h); where j = 0h to Fh

Figure 20-46. OUTPUTXBARG7SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-61. OUTPUTXBARG7SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-61. OUTPUTXBARG7SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-61. OUTPUTXBARG7SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.5.15 OUTPUTXBARG8SEL_j Register (Offset = 120h + formula) [Reset = 00000000h]

OUTPUTXBARG8SEL_j is shown in Figure 20-47 and described in Table 20-62.

Return to the Summary Table.

OUTPUTXBAR G8 Input Select

Offset = 120h + (j * 40h); where j = 0h to Fh

Figure 20-47. OUTPUTXBARG8SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-62. OUTPUTXBARG8SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-62. OUTPUTXBARG8SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-62. OUTPUTXBARG8SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.5.16 OUTPUTXBARG9SEL_j Register (Offset = 124h + formula) [Reset = 00000000h]

OUTPUTXBARG9SEL_j is shown in Figure 20-48 and described in Table 20-63.

Return to the Summary Table.

OUTPUTXBAR G9 Input Select

Offset = 124h + (j * 40h); where j = 0h to Fh

Figure 20-48. OUTPUTXBARG9SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-63. OUTPUTXBARG9SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-63. OUTPUTXBARG9SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-63. OUTPUTXBARG9SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.6 MDL_XBAR_REGS Registers

Table 20-64 lists the memory-mapped registers for the MDL_XBAR_REGS registers. All register offset addresses not listed in Table 20-64 should be considered as reserved locations and the register contents should not be modified.

Table 20-64. MDL_XBAR_REGS Registers

Offset	Acronym	Register Name	Protection
30h	MDLXBAROutInvert	Output Signal Invert Select	
80h	MDLXBARLock	Configuration Lock register	
100h + formula	MDLXBARG0SEL_j	MDLXBAR G0 Input Select	
104h + formula	MDLXBARG1SEL_j	MDLXBAR G1 Input Select	
108h + formula	MDLXBARG2SEL_j	MDLXBAR G2 Input Select	

Complex bit access types are encoded to fit into small table cells. Table 20-65 shows the codes that are used for access types in this section.

Table 20-65. MDL_XBAR_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

20.5.6.1 MDLXBAROutInvert Register (Offset = 30h) [Reset = 00000000h]

MDLXBAROutInvert is shown in Figure 20-49 and described in Table 20-66.

Return to the Summary Table.

Output Signal Invert Select

Figure 20-49. MDLXBAROutInvert Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
XBAR16	XBAR15	XBAR14	XBAR13	XBAR12	XBAR11	XBAR10	XBAR9
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
XBAR8	XBAR7	XBAR6	XBAR5	XBAR4	XBAR3	XBAR2	XBAR1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-66. MDLXBAROutInvert Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	XBAR16	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
14	XBAR15	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
13	XBAR14	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
12	XBAR13	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
11	XBAR12	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
10	XBAR11	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
9	XBAR10	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn

Table 20-66. MDLXBAROutInvert Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	XBAR9	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
7	XBAR8	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
6	XBAR7	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
5	XBAR6	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
4	XBAR5	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
3	XBAR4	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
2	XBAR3	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
1	XBAR2	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
0	XBAR1	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn

20.5.6.2 MDLXBARLock Register (Offset = 80h) [Reset = 00000000h]

MDLXBARLock is shown in Figure 20-50 and described in Table 20-67.

Return to the Summary Table.

Configuration Lock register

Figure 20-50. MDLXBARLock Register

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

Table 20-67. MDLXBARLock Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Bit-0 of this register can be set only if KEY= 0x5a5a Reset type: XRSn
15-1	RESERVED	R-0	0h	Reserved
0	LOCK	R/WSonce	0h	Locks the configuration for EMDL-XBAR. Once the configuration is locked, writes to the below registers for EMDL-XBAR is blocked. Registers Affected by the LOCK mechanism: MDLXBAR Gx SEL MDLXBAR.Invert 0: Writes to the above registers are allowed 1: Writes to the above registers are blocked Note: [1] LOCK mechanism only apples to writes. Reads are never blocked. Reset type: XRSn

20.5.6.3 MDLXBARG0SEL_j Register (Offset = 100h + formula) [Reset = 00000000h]

MDLXBARG0SEL_j is shown in Figure 20-51 and described in Table 20-68.

Return to the Summary Table.

MDLXBAR G0 Input Select

Offset = 100h + (j * 40h); where j = 0h to Fh

Figure 20-51. MDLXBARG0SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-68. MDLXBARG0SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-68. MDLXBARG0SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-68. MDLXBARG0SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.6.4 MDLXBARG1SEL_j Register (Offset = 104h + formula) [Reset = 00000000h]

MDLXBARG1SEL_j is shown in Figure 20-52 and described in Table 20-69.

Return to the Summary Table.

MDLXBAR G1 Input Select

Offset = 104h + (j * 40h); where j = 0h to Fh

Figure 20-52. MDLXBARG1SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-69. MDLXBARG1SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-69. MDLXBARG1SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-69. MDLXBARG1SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.6.5 MDLXBARG2SEL_j Register (Offset = 108h + formula) [Reset = 00000000h]

MDLXBARG2SEL_j is shown in Figure 20-53 and described in Table 20-70.

Return to the Summary Table.

MDLXBAR G2 Input Select

Offset = 108h + (j * 40h); where j = 0h to Fh

Figure 20-53. MDLXBARG2SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-70. MDLXBARG2SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-70. MDLXBARG2SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-70. MDLXBARG2SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.7 ICL_XBAR_REGS Registers

Table 20-71 lists the memory-mapped registers for the ICL_XBAR_REGS registers. All register offset addresses not listed in Table 20-71 should be considered as reserved locations and the register contents should not be modified.

Table 20-71. ICL_XBAR_REGS Registers

Offset	Acronym	Register Name	Protection
30h	ICLXBAROutInvert	Output Signal Invert Select	
80h	ICLXBARLock	Configuration Lock register	
100h + formula	ICLXBARG0SEL_j	ICLXBAR G0 Input Select	
104h + formula	ICLXBARG1SEL_j	ICLXBAR G1 Input Select	
108h + formula	ICLXBARG2SEL_j	ICLXBAR G2 Input Select	

Complex bit access types are encoded to fit into small table cells. Table 20-72 shows the codes that are used for access types in this section.

Table 20-72. ICL_XBAR_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

20.5.7.1 ICLXBAROutInvert Register (Offset = 30h) [Reset = 00000000h]

ICLXBAROutInvert is shown in Figure 20-54 and described in Table 20-73.

Return to the Summary Table.

Output Signal Invert Select

Figure 20-54. ICLXBAROutInvert Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
XBAR16	XBAR15	XBAR14	XBAR13	XBAR12	XBAR11	XBAR10	XBAR9
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
XBAR8	XBAR7	XBAR6	XBAR5	XBAR4	XBAR3	XBAR2	XBAR1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-73. ICLXBAROutInvert Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	XBAR16	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
14	XBAR15	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
13	XBAR14	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
12	XBAR13	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
11	XBAR12	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
10	XBAR11	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
9	XBAR10	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn

Table 20-73. ICLXBAROutInvert Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	XBAR9	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
7	XBAR8	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
6	XBAR7	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
5	XBAR6	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
4	XBAR5	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
3	XBAR4	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
2	XBAR3	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
1	XBAR2	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
0	XBAR1	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn

20.5.7.2 ICLXBARLock Register (Offset = 80h) [Reset = 00000000h]

ICLXBARLock is shown in Figure 20-55 and described in Table 20-74.

Return to the Summary Table.

Configuration Lock register

Figure 20-55. ICLXBARLock Register

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

Table 20-74. ICLXBARLock Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Bit-0 of this register can be set only if KEY= 0x5a5a Reset type: XRSn
15-1	RESERVED	R-0	0h	Reserved
0	LOCK	R/WSonce	0h	Locks the configuration for EICL-XBAR. Once the configuration is locked, writes to the below registers for EICL-XBAR is blocked. Registers Affected by the LOCK mechanism: ICLXBAR Gx SEL ICLXBAR.Invert 0: Writes to the above registers are allowed 1: Writes to the above registers are blocked Note: [1] LOCK mechanism only apples to writes. Reads are never blocked. Reset type: XRSn

20.5.7.3 ICLXBARG0SEL_j Register (Offset = 100h + formula) [Reset = 00000000h]

ICLXBARG0SEL_j is shown in Figure 20-56 and described in Table 20-75.

Return to the Summary Table.

ICLXBAR G0 Input Select

Offset = 100h + (j * 40h); where j = 0h to Fh

Figure 20-56. ICLXBARG0SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-75. ICLXBARG0SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-75. ICLXBARG0SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-75. ICLXBARG0SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.7.4 ICLXBARG1SEL_j Register (Offset = 104h + formula) [Reset = 00000000h]

ICLXBARG1SEL_j is shown in Figure 20-57 and described in Table 20-76.

Return to the Summary Table.

ICLXBAR G1 Input Select

Offset = 104h + (j * 40h); where j = 0h to Fh

Figure 20-57. ICLXBARG1SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-76. ICLXBARG1SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-76. ICLXBARG1SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-76. ICLXBARG1SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.7.5 ICLXBARG2SEL_j Register (Offset = 108h + formula) [Reset = 00000000h]

ICLXBARG2SEL_j is shown in Figure 20-58 and described in Table 20-77.

Return to the Summary Table.

ICLXBAR G2 Input Select

Offset = 108h + (j * 40h); where j = 0h to Fh

Figure 20-58. ICLXBARG2SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-77. ICLXBARG2SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-77. ICLXBARG2SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 20-77. ICLXBARG2SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

20.5.8 OUTPUTXBAR_FLAG_REGS Registers

Table 20-78 lists the memory-mapped registers for the OUTPUTXBAR_FLAG_REGS registers. All register offset addresses not listed in Table 20-78 should be considered as reserved locations and the register contents should not be modified.

Table 20-78. OUTPUTXBAR_FLAG_REGS Registers

Offset	Acronym	Register Name	Protection
10h	OUTPUTXBARStatus	Output Signal Status register	
18h	OUTPUTXBARFlag	Output latched flag register	
1Ch	OUTPUTXBARFlagClear	Output latched flag clear register	
20h	OUTPUTXBARFlagForce	Output latched flag Force register	

Complex bit access types are encoded to fit into small table cells. Table 20-79 shows the codes that are used for access types in this section.

Table 20-79. OUTPUTXBAR_FLAG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

20.5.8.1 OUTPUTXBARStatus Register (Offset = 10h) [Reset = 00000000h]

OUTPUTXBARStatus is shown in Figure 20-59 and described in Table 20-80.

Return to the Summary Table.

Output Signal Status register

Figure 20-59. OUTPUTXBARStatus Register

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

Table 20-80. OUTPUTXBARStatus Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	STS	R	0h	Output Signal Status: 0 Low State 1 High State Reset type: XRSn

20.5.8.2 OUTPUTXBARFlag Register (Offset = 18h) [Reset = 00000000h]

OUTPUTXBARFlag is shown in Figure 20-60 and described in Table 20-81.

Return to the Summary Table.

Output latched flag register

Figure 20-60. OUTPUTXBARFlag Register

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

Table 20-81. OUTPUTXBARFlag Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	FLG	R	0h	Output Signal Latched Flag: 0 No Input Latched Event 1 Latched Event Reset type: XRSn

20.5.8.3 OUTPUTXBARFlagClear Register (Offset = 1Ch) [Reset = 00000000h]

OUTPUTXBARFlagClear is shown in Figure 20-61 and described in Table 20-82.

Return to the Summary Table.

Output latched flag clear register

Figure 20-61. OUTPUTXBARFlagClear Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															FLG
R-0-0h															R-0/W1S-0h

Table 20-82. OUTPUTXBARFlagClear Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	FLG	R-0/W1S	0h	Output Signal Latched Flag Clear: Write of 1 will clear flag. If Flag is being cleared by S/W and a new flag is being latched, hardware has priority. Reset type: XRSn

20.5.8.4 OUTPUTXBARFlagForce Register (Offset = 20h) [Reset = 00000000h]

OUTPUTXBARFlagForce is shown in Figure 20-62 and described in Table 20-83.

Return to the Summary Table.

Output latched flag Force register

Figure 20-62. OUTPUTXBARFlagForce Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															FLG
R-0-0h															R-0/W1S-0h

Table 20-83. OUTPUTXBARFlagForce Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	FLG	R-0/W1S	0h	Output Signal Latched Flag Force: Write of 1 will set flag. Reset type: XRSn

20.5.9 XBAR_REGS Registers

Table 20-84 lists the memory-mapped registers for the XBAR_REGS registers. All register offset addresses not listed in Table 20-84 should be considered as reserved locations and the register contents should not be modified.

Table 20-84. XBAR_REGS Registers

Offset	Acronym	Register Name	Protection
0h	XBARFLG1	X-Bar Input Flag Register 1	
4h	XBARFLG2	X-Bar Input Flag Register 2	
8h	XBARFLG3	X-Bar Input Flag Register 3	
Ch	XBARFLG4	X-Bar Input Flag Register 4	
10h	XBARFLG5	X-Bar Input Flag Register 5	
14h	XBARFLG6	X-Bar Input Flag Register 6	
18h	XBARFLG7	X-Bar Input Flag Register 7	
1Ch	XBARFLG8	X-Bar Input Flag Register 8	
20h	XBARFLG9	X-Bar Input Flag Register 9	
24h	XBARFLG10	X-Bar Input Flag Register 10	
28h	XBARFLG11	X-Bar Input Flag Register 11	
2Ch	XBARFLG12	X-Bar Input Flag Register 12	
30h	XBARFLG13	X-Bar Input Flag Register 13	
34h	XBARFLG14	X-Bar Input Flag Register 14	
38h	XBARFLG15	X-Bar Input Flag Register 15	
3Ch	XBARFLG16	X-Bar Input Flag Register 16	
40h	XBARFLG17	X-Bar Input Flag Register 16	
44h	XBARFLG18	X-Bar Input Flag Register 16	
100h	XBARCLR1	X-Bar Input Flag Clear Register 1	
104h	XBARCLR2	X-Bar Input Flag Clear Register 2	
108h	XBARCLR3	X-Bar Input Flag Clear Register 3	
10Ch	XBARCLR4	X-Bar Input Flag Clear Register 4	
110h	XBARCLR5	X-Bar Input Flag Clear Register 5	
114h	XBARCLR6	X-Bar Input Flag Clear Register 6	
118h	XBARCLR7	X-Bar Input Flag Clear Register 7	
11Ch	XBARCLR8	X-Bar Input Flag Clear Register 8	
120h	XBARCLR9	X-Bar Input Flag Clear Register 9	
124h	XBARCLR10	X-Bar Input Flag Clear Register 10	
128h	XBARCLR11	X-Bar Input Flag Clear Register 11	
12Ch	XBARCLR12	X-Bar Input Flag Clear Register 12	
130h	XBARCLR13	X-Bar Input Flag Clear Register 13	
134h	XBARCLR14	X-Bar Input Flag Clear Register 14	
138h	XBARCLR15	X-Bar Input Flag Clear Register 15	
13Ch	XBARCLR16	X-Bar Input Flag Clear Register 15	
140h	XBARCLR17	X-Bar Input Flag Clear Register 16	
144h	XBARCLR18	X-Bar Input Flag Clear Register 16	

Complex bit access types are encoded to fit into small table cells. Table 20-85 shows the codes that are used for access types in this section.

Table 20-85. XBAR_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

20.5.9.1 XBARFLG1 Register (Offset = 0h) [Reset = 00000000h]

XBARFLG1 is shown in Figure 20-63 and described in Table 20-86.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-63. XBARFLG1 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
CMPSS12_CTRIPL	CMPSS12_CTRIPH	CMPSS11_CTRIPL	CMPSS11_CTRIPH	CMPSS10_CTRIPL	CMPSS10_CTRIPH	CMPSS9_CTRIPL	CMPSS9_CTRIPH
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
CMPSS8_CTRIPH	CMPSS8_CTRIPL	CMPSS7_CTRIPH	CMPSS7_CTRIPL	CMPSS6_CTRIPH	CMPSS6_CTRIPL	CMPSS5_CTRIPH	CMPSS5_CTRIPL
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
CMPSS4_CTRIPH	CMPSS4_CTRIPL	CMPSS3_CTRIPH	CMPSS3_CTRIPL	CMPSS2_CTRIPH	CMPSS2_CTRIPL	CMPSS1_CTRIPH	CMPSS1_CTRIPL
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-86. XBARFLG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	RESERVED	R	0h	Reserved
29	RESERVED	R	0h	Reserved
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	RESERVED	R	0h	Reserved
24	RESERVED	R	0h	Reserved
23	CMPSS12_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS12_CTRIPL input was triggered 0: CMPSS12_CTRIPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
22	CMPSS12_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS12_CTRIPH input was triggered 0: CMPSS12_CTRIPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-86. XBARFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	CMPSS11_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS11_CTRIPL input was triggered 0: CMPSS11_CTRIPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
20	CMPSS11_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS11_CTRIPH input was triggered 0: CMPSS11_CTRIPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
19	CMPSS10_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS10_CTRIPL input was triggered 0: CMPSS10_CTRIPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
18	CMPSS10_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS10_CTRIPH input was triggered 0: CMPSS10_CTRIPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
17	CMPSS9_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS9_CTRIPL input was triggered 0: CMPSS9_CTRIPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
16	CMPSS9_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS9_CTRIPH input was triggered 0: CMPSS9_CTRIPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	CMPSS8_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS8_CTRIPH input was triggered 0: CMPSS8_CTRIPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	CMPSS8_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS8_CTRIPL input was triggered 0: CMPSS8_CTRIPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-86. XBARFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	CMPSS7_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS7_CTRIPH input was triggered 0: CMPSS7_CTRIPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
12	CMPSS7_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS7_CTRIPL input was triggered 0: CMPSS7_CTRIPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
11	CMPSS6_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS6_CTRIPH input was triggered 0: CMPSS6_CTRIPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	CMPSS6_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS6_CTRIPL input was triggered 0: CMPSS6_CTRIPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	CMPSS5_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS5_CTRIPH input was triggered 0: CMPSS5_CTRIPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
8	CMPSS5_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS5_CTRIPL input was triggered 0: CMPSS5_CTRIPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	CMPSS4_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS4_CTRIPH input was triggered 0: CMPSS4_CTRIPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	CMPSS4_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS4_CTRIPL input was triggered 0: CMPSS4_CTRIPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-86. XBARFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	CMPSS3_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS3_CTRIPH input was triggered 0: CMPSS3_CTRIPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	CMPSS3_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS3_CTRIPL input was triggered 0: CMPSS3_CTRIPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
3	CMPSS2_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS2_CTRIPH input was triggered 0: CMPSS2_CTRIPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	CMPSS2_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS2_CTRIPL input was triggered 0: CMPSS2_CTRIPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	CMPSS1_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS1_CTRIPH input was triggered 0: CMPSS1_CTRIPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	CMPSS1_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS1_CTRIPL input was triggered 0: CMPSS1_CTRIPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.2 XBARFLG2 Register (Offset = 4h) [Reset = 00000000h]

XBARFLG2 is shown in Figure 20-64 and described in Table 20-87.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-64. XBARFLG2 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
CMPSS12_CTRIPOUTL	CMPSS12_CTRIPOUTH	CMPSS11_CTRIPOUTL	CMPSS11_CTRIPOUTH	CMPSS10_CTRIPOUTL	CMPSS10_CTRIPOUTH	CMPSS9_CTRIPOUTL	CMPSS9_CTRIPOUTH
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
CMPSS8_CTRIPOUTH	CMPSS8_CTRIPOUTL	CMPSS7_CTRIPOUTH	CMPSS7_CTRIPOUTL	CMPSS6_CTRIPOUTH	CMPSS6_CTRIPOUTL	CMPSS5_CTRIPOUTH	CMPSS5_CTRIPOUTL
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
CMPSS4_CTRIPOUTH	CMPSS4_CTRIPOUTL	CMPSS3_CTRIPOUTH	CMPSS3_CTRIPOUTL	CMPSS2_CTRIPOUTH	CMPSS2_CTRIPOUTL	CMPSS1_CTRIPOUTH	CMPSS1_CTRIPOUTL
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-87. XBARFLG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	RESERVED	R	0h	Reserved
29	RESERVED	R	0h	Reserved
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	RESERVED	R	0h	Reserved
24	RESERVED	R	0h	Reserved
23	CMPSS12_CTRIPOUTL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS12_CTRIPOUTL input was triggered 0: CMPSS12_CTRIPOUTL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
22	CMPSS12_CTRIPOUTH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS12_CTRIPOUTH input was triggered 0: CMPSS12_CTRIPOUTH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-87. XBARFLG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	CMPSS11_CTRIPOUTL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS11_CTRIPOUTL input was triggered 0: CMPSS11_CTRIPOUTL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
20	CMPSS11_CTRIPOUTH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS11_CTRIPOUTH input was triggered 0: CMPSS11_CTRIPOUTH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
19	CMPSS10_CTRIPOUTL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS10_CTRIPOUTL input was triggered 0: CMPSS10_CTRIPOUTL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
18	CMPSS10_CTRIPOUTH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS10_CTRIPOUTH input was triggered 0: CMPSS10_CTRIPOUTH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
17	CMPSS9_CTRIPOUTL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS9_CTRIPOUTL input was triggered 0: CMPSS9_CTRIPOUTL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
16	CMPSS9_CTRIPOUTH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS9_CTRIPOUTH input was triggered 0: CMPSS9_CTRIPOUTH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	CMPSS8_CTRIPOUTH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS8_CTRIPOUTH input was triggered 0: CMPSS8_CTRIPOUTH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	CMPSS8_CTRIPOUTL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS8_CTRIPOUTL input was triggered 0: CMPSS8_CTRIPOUTL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-87. XBARFLG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	CMPSS7_CTRIPOUTH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS7_CTRIPOUTH input was triggered 0: CMPSS7_CTRIPOUTH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
12	CMPSS7_CTRIPOUTL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS7_CTRIPOUTL input was triggered 0: CMPSS7_CTRIPOUTL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
11	CMPSS6_CTRIPOUTH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS6_CTRIPOUTH input was triggered 0: CMPSS6_CTRIPOUTH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	CMPSS6_CTRIPOUTL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS6_CTRIPOUTL input was triggered 0: CMPSS6_CTRIPOUTL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	CMPSS5_CTRIPOUTH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS5_CTRIPOUTH input was triggered 0: CMPSS5_CTRIPOUTH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
8	CMPSS5_CTRIPOUTL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS5_CTRIPOUTL input was triggered 0: CMPSS5_CTRIPOUTL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	CMPSS4_CTRIPOUTH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS4_CTRIPOUTH input was triggered 0: CMPSS4_CTRIPOUTH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	CMPSS4_CTRIPOUTL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS4_CTRIPOUTL input was triggered 0: CMPSS4_CTRIPOUTL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-87. XBARFLG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	CMPSS3_CTRIPOUTH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS3_CTRIPOUTH input was triggered 0: CMPSS3_CTRIPOUTH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	CMPSS3_CTRIPOUTL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS3_CTRIPOUTL input was triggered 0: CMPSS3_CTRIPOUTL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
3	CMPSS2_CTRIPOUTH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS2_CTRIPOUTH input was triggered 0: CMPSS2_CTRIPOUTH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	CMPSS2_CTRIPOUTL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS2_CTRIPOUTL input was triggered 0: CMPSS2_CTRIPOUTL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	CMPSS1_CTRIPOUTH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS1_CTRIPOUTH input was triggered 0: CMPSS1_CTRIPOUTH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	CMPSS1_CTRIPOUTL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CMPSS1_CTRIPOUTL input was triggered 0: CMPSS1_CTRIPOUTL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.3 XBARFLG3 Register (Offset = 8h) [Reset = 00000000h]

XBARFLG3 is shown in Figure 20-65 and described in Table 20-88.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-65. XBARFLG3 Register

31	30	29	28	27	26	25	24
SD4FLT4_COMPL	SD4FLT4_COMPH	SD4FLT3_COMPL	SD4FLT3_COMPH	SD4FLT2_COMPL	SD4FLT2_COMPH	SD4FLT1_COMPL	SD4FLT1_COMPH
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
SD3FLT4_COMPL	SD3FLT4_COMPH	SD3FLT3_COMPL	SD3FLT3_COMPH	SD3FLT2_COMPL	SD3FLT2_COMPH	SD3FLT1_COMPL	SD3FLT1_COMPH
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
SD2FLT4_COMPH	SD2FLT4_COMPL	SD2FLT3_COMPH	SD2FLT3_COMPL	SD2FLT2_COMPH	SD2FLT2_COMPL	SD2FLT1_COMPH	SD2FLT1_COMPL
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
SD1FLT4_COMPH	SD1FLT4_COMPL	SD1FLT3_COMPH	SD1FLT3_COMPL	SD1FLT2_COMPH	SD1FLT2_COMPL	SD1FLT1_COMPH	SD1FLT1_COMPL
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-88. XBARFLG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SD4FLT4_COMPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD4FLT4_COMPL input was triggered 0: SD4FLT4_COMPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
30	SD4FLT4_COMPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD4FLT4_COMPH input was triggered 0: SD4FLT4_COMPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
29	SD4FLT3_COMPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD4FLT3_COMPL input was triggered 0: SD4FLT3_COMPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-88. XBARFLG3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	SD4FLT3_COMPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD4FLT3_COMPH input was triggered 0: SD4FLT3_COMPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
27	SD4FLT2_COMPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD4FLT2_COMPL input was triggered 0: SD4FLT2_COMPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
26	SD4FLT2_COMPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD4FLT2_COMPH input was triggered 0: SD4FLT2_COMPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
25	SD4FLT1_COMPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD4FLT1_COMPL input was triggered 0: SD4FLT1_COMPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
24	SD4FLT1_COMPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD4FLT1_COMPH input was triggered 0: SD4FLT1_COMPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
23	SD3FLT4_COMPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD3FLT4_COMPL input was triggered 0: SD3FLT4_COMPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
22	SD3FLT4_COMPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD3FLT4_COMPH input was triggered 0: SD3FLT4_COMPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
21	SD3FLT3_COMPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD3FLT3_COMPL input was triggered 0: SD3FLT3_COMPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-88. XBARFLG3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	SD3FLT3_COMPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD3FLT3_COMPH input was triggered 0: SD3FLT3_COMPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
19	SD3FLT2_COMPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD3FLT2_COMPL input was triggered 0: SD3FLT2_COMPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
18	SD3FLT2_COMPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD3FLT2_COMPH input was triggered 0: SD3FLT2_COMPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
17	SD3FLT1_COMPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD3FLT1_COMPL input was triggered 0: SD3FLT1_COMPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
16	SD3FLT1_COMPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD3FLT1_COMPH input was triggered 0: SD3FLT1_COMPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	SD2FLT4_COMPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD2FLT4_COMPH input was triggered 0: SD2FLT4_COMPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	SD2FLT4_COMPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD2FLT4_COMPL input was triggered 0: SD2FLT4_COMPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
13	SD2FLT3_COMPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD2FLT3_COMPH input was triggered 0: SD2FLT3_COMPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-88. XBARFLG3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	SD2FLT3_COMPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD2FLT3_COMPL input was triggered 0: SD2FLT3_COMPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
11	SD2FLT2_COMPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD2FLT2_COMPH input was triggered 0: SD2FLT2_COMPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	SD2FLT2_COMPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD2FLT2_COMPL input was triggered 0: SD2FLT2_COMPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	SD2FLT1_COMPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD2FLT1_COMPH input was triggered 0: SD2FLT1_COMPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
8	SD2FLT1_COMPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD2FLT1_COMPL input was triggered 0: SD2FLT1_COMPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	SD1FLT4_COMPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD1FLT4_COMPH input was triggered 0: SD1FLT4_COMPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	SD1FLT4_COMPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD1FLT4_COMPL input was triggered 0: SD1FLT4_COMPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	SD1FLT3_COMPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD1FLT3_COMPH input was triggered 0: SD1FLT3_COMPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-88. XBARFLG3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	SD1FLT3_COMPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD1FLT3_COMPL input was triggered 0: SD1FLT3_COMPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
3	SD1FLT2_COMPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD1FLT2_COMPH input was triggered 0: SD1FLT2_COMPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	SD1FLT2_COMPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD1FLT2_COMPL input was triggered 0: SD1FLT2_COMPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	SD1FLT1_COMPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD1FLT1_COMPH input was triggered 0: SD1FLT1_COMPH Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	SD1FLT1_COMPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: SD1FLT1_COMPL input was triggered 0: SD1FLT1_COMPL Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.4 XBARFLG4 Register (Offset = Ch) [Reset = 00000000h]

XBARFLG4 is shown in Figure 20-66 and described in Table 20-89.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-66. XBARFLG4 Register

31	30	29	28	27	26	25	24
INPUT32	INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
INPUT24	INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
INPUT16	INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
INPUT8	INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-89. XBARFLG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT32	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT32 input was triggered 0: INPUT32 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
30	INPUT31	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT31 input was triggered 0: INPUT31 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
29	INPUT30	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT30 input was triggered 0: INPUT30 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
28	INPUT29	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT29 input was triggered 0: INPUT29 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-89. XBARFLG4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	INPUT28	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT28 input was triggered 0: INPUT28 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
26	INPUT27	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT27 input was triggered 0: INPUT27 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
25	INPUT26	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT26 input was triggered 0: INPUT26 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
24	INPUT25	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT25 input was triggered 0: INPUT25 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
23	INPUT24	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT24 input was triggered 0: INPUT24 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
22	INPUT23	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT23 input was triggered 0: INPUT23 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
21	INPUT22	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT22 input was triggered 0: INPUT22 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
20	INPUT21	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT21 input was triggered 0: INPUT21 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-89. XBARFLG4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	INPUT20	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT20 input was triggered 0: INPUT20 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
18	INPUT19	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT19 input was triggered 0: INPUT19 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
17	INPUT18	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT18 input was triggered 0: INPUT18 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
16	INPUT17	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT17 input was triggered 0: INPUT17 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	INPUT16	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT16 input was triggered 0: INPUT16 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	INPUT15	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT15 input was triggered 0: INPUT15 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
13	INPUT14	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT14 input was triggered 0: INPUT14 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
12	INPUT13	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT13 input was triggered 0: INPUT13 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-89. XBARFLG4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	INPUT12	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT12 input was triggered 0: INPUT12 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	INPUT11	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT11 input was triggered 0: INPUT11 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	INPUT10	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT10 input was triggered 0: INPUT10 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
8	INPUT9	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT9 input was triggered 0: INPUT9 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	INPUT8	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT8 input was triggered 0: INPUT8 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	INPUT7	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT7 input was triggered 0: INPUT7 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	INPUT6	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT6 input was triggered 0: INPUT6 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	INPUT5	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT5 input was triggered 0: INPUT5 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-89. XBARFLG4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	INPUT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT4 input was triggered 0: INPUT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	INPUT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT3 input was triggered 0: INPUT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	INPUT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT2 input was triggered 0: INPUT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	INPUT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: INPUT1 input was triggered 0: INPUT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.5 XBARFLG5 Register (Offset = 10h) [Reset = 00000000h]

XBARFLG5 is shown in Figure 20-67 and described in Table 20-90.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-67. XBARFLG5 Register

31	30	29	28	27	26	25	24
CPU3_ADCCHECKEVT4	CPU3_ADCCHECKEVT3	CPU3_ADCCHECKEVT2	CPU3_ADCCHECKEVT1	CPU2_ADCCHECKEVT4	CPU2_ADCCHECKEVT3	CPU2_ADCCHECKEVT2	CPU2_ADCCHECKEVT1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
CPU1_ADCCHECKEVT4	CPU1_ADCCHECKEVT3	CPU1_ADCCHECKEVT2	CPU1_ADCCHECKEVT1	ADCEEVT4	ADCEEVT3	ADCEEVT2	ADCEEVT1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
ADCDEVT4	ADCDEVT3	ADCDEVT2	ADCDEVT1	ADCCEVT4	ADCCEVT3	ADCCEVT2	ADCCEVT1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
ADCBEVT4	ADCBEVT3	ADCBEVT2	ADCBEVT1	ADCAEVT4	ADCAEVT3	ADCAEVT2	ADCAEVT1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-90. XBARFLG5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CPU3_ADCCHECKEVT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU3_ADCCHECKEVT4 input was triggered 0: CPU3_ADCCHECKEVT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
30	CPU3_ADCCHECKEVT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU3_ADCCHECKEVT3 input was triggered 0: CPU3_ADCCHECKEVT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
29	CPU3_ADCCHECKEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU3_ADCCHECKEVT2 input was triggered 0: CPU3_ADCCHECKEVT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
28	CPU3_ADCCHECKEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU3_ADCCHECKEVT1 input was triggered 0: CPU3_ADCCHECKEVT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-90. XBARFLG5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	CPU2_ADCCHECKEVT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU2_ADCCHECKEVT4 input was triggered 0: CPU2_ADCCHECKEVT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
26	CPU2_ADCCHECKEVT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU2_ADCCHECKEVT3 input was triggered 0: CPU2_ADCCHECKEVT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
25	CPU2_ADCCHECKEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU2_ADCCHECKEVT2 input was triggered 0: CPU2_ADCCHECKEVT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
24	CPU2_ADCCHECKEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU2_ADCCHECKEVT1 input was triggered 0: CPU2_ADCCHECKEVT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
23	CPU1_ADCCHECKEVT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU1_ADCCHECKEVT4 input was triggered 0: CPU1_ADCCHECKEVT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
22	CPU1_ADCCHECKEVT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU1_ADCCHECKEVT3 input was triggered 0: CPU1_ADCCHECKEVT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
21	CPU1_ADCCHECKEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU1_ADCCHECKEVT2 input was triggered 0: CPU1_ADCCHECKEVT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
20	CPU1_ADCCHECKEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU1_ADCCHECKEVT1 input was triggered 0: CPU1_ADCCHECKEVT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-90. XBARFLG5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	ADCEEVT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCEEVT4 input was triggered 0: ADCEEVT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
18	ADCEEVT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCEEVT3 input was triggered 0: ADCEEVT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
17	ADCEEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCEEVT2 input was triggered 0: ADCEEVT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
16	ADCEEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCEEVT1 input was triggered 0: ADCEEVT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	ADCDEVT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCDEVT4 input was triggered 0: ADCDEVT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	ADCDEVT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCDEVT3 input was triggered 0: ADCDEVT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
13	ADCDEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCDEVT2 input was triggered 0: ADCDEVT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
12	ADCDEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCDEVT1 input was triggered 0: ADCDEVT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-90. XBARFLG5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	ADCCEVT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCCEVT4 input was triggered 0: ADCCEVT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	ADCCEVT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCCEVT3 input was triggered 0: ADCCEVT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	ADCCEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCCEVT2 input was triggered 0: ADCCEVT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
8	ADCCEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCCEVT1 input was triggered 0: ADCCEVT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	ADCBEVT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCBEVT4 input was triggered 0: ADCBEVT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	ADCBEVT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCBEVT3 input was triggered 0: ADCBEVT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	ADCBEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCBEVT2 input was triggered 0: ADCBEVT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	ADCBEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCBEVT1 input was triggered 0: ADCBEVT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-90. XBARFLG5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	ADCAEVT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCAEVT4 input was triggered 0: ADCAEVT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	ADCAEVT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCAEVT3 input was triggered 0: ADCAEVT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	ADCAEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCAEVT2 input was triggered 0: ADCAEVT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	ADCAEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCAEVT1 input was triggered 0: ADCAEVT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.6 XBARFLG6 Register (Offset = 14h) [Reset = 00000000h]

XBARFLG6 is shown in Figure 20-68 and described in Table 20-91.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-68. XBARFLG6 Register

31	30	29	28	27	26	25	24
ECATSYNC1	ECATSYNC0	WADI1OUT7	WADI1OUT6	WADI1OUT5	WADI1OUT4	WADI1OUT3	WADI1OUT2
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
WADI1OUT1	WADI1OUT0	RESERVED	RESERVED	FSID_RX_TRIG 1	FSIC_RX_TRIG 1	FSIB_RX_TRIG 1	FSIA_RX_TRIG 1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	CLB6_OUT5	CLB6_OUT4	CLB5_OUT5	CLB5_OUT4
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
CLB4_OUT5	CLB4_OUT4	CLB3_OUT5	CLB3_OUT4	CLB2_OUT5	CLB2_OUT4	CLB1_OUT5	CLB1_OUT4
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-91. XBARFLG6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ECATSYNC1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ECATSYNC1 input was triggered 0: ECATSYNC1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
30	ECATSYNC0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ECATSYNC0 input was triggered 0: ECATSYNC0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
29	WADI1OUT7	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: WADI1OUT7 input was triggered 0: WADI1OUT7 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
28	WADI1OUT6	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: WADI1OUT6 input was triggered 0: WADI1OUT6 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-91. XBARFLG6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	WADI1OUT5	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: WADI1OUT5 input was triggered 0: WADI1OUT5 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
26	WADI1OUT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: WADI1OUT4 input was triggered 0: WADI1OUT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
25	WADI1OUT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: WADI1OUT3 input was triggered 0: WADI1OUT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
24	WADI1OUT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: WADI1OUT2 input was triggered 0: WADI1OUT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
23	WADI1OUT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: WADI1OUT1 input was triggered 0: WADI1OUT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
22	WADI1OUT0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: WADI1OUT0 input was triggered 0: WADI1OUT0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
21	RESERVED	R	0h	Reserved
20	RESERVED	R	0h	Reserved
19	FSID_RX_TRIG1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSID_RX_TRIG1 input was triggered 0: FSID_RX_TRIG1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
18	FSIC_RX_TRIG1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIC_RX_TRIG1 input was triggered 0: FSIC_RX_TRIG1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-91. XBARFLG6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	FSIB_RX_TRIG1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIB_RX_TRIG1 input was triggered 0: FSIB_RX_TRIG1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
16	FSIA_RX_TRIG1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIA_RX_TRIG1 input was triggered 0: FSIA_RX_TRIG1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	CLB6_OUT5	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB6_OUT5 input was triggered 0: CLB6_OUT5 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	CLB6_OUT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB6_OUT4 input was triggered 0: CLB6_OUT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	CLB5_OUT5	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB5_OUT5 input was triggered 0: CLB5_OUT5 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
8	CLB5_OUT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB5_OUT4 input was triggered 0: CLB5_OUT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	CLB4_OUT5	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB4_OUT5 input was triggered 0: CLB4_OUT5 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-91. XBARFLG6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	CLB4_OUT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB4_OUT4 input was triggered 0: CLB4_OUT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	CLB3_OUT5	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB3_OUT5 input was triggered 0: CLB3_OUT5 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	CLB3_OUT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB3_OUT4 input was triggered 0: CLB3_OUT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
3	CLB2_OUT5	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB2_OUT5 input was triggered 0: CLB2_OUT5 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	CLB2_OUT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB2_OUT4 input was triggered 0: CLB2_OUT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	CLB1_OUT5	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB1_OUT5 input was triggered 0: CLB1_OUT5 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	CLB1_OUT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB1_OUT4 input was triggered 0: CLB1_OUT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.7 XBARFLG7 Register (Offset = 18h) [Reset = 00000000h]

XBARFLG7 is shown in Figure 20-69 and described in Table 20-92.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-69. XBARFLG7 Register

31	30	29	28	27	26	25	24
WADI2OUT7	WADI2OUT6	WADI2OUT5	WADI2OUT4	WADI2OUT3	WADI2OUT2	WADI2OUT1	WADI2OUT0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	EXTSYNCOUT	ESMGENEVT	ADCSOCB	ADCSOCA
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	ECAP6_TRIPOUT	ECAP5_TRIPOUT	ECAP4_TRIPOUT	ECAP3_TRIPOUT	ECAP2_TRIPOUT	ECAP1_TRIPOUT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	ECAP6_OUT	ECAP5_OUT	ECAP4_OUT	ECAP3_OUT	ECAP2_OUT	ECAP1_OUT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-92. XBARFLG7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	WADI2OUT7	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: WADI2OUT7 input was triggered 0: WADI2OUT7 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
30	WADI2OUT6	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: WADI2OUT6 input was triggered 0: WADI2OUT6 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
29	WADI2OUT5	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: WADI2OUT5 input was triggered 0: WADI2OUT5 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
28	WADI2OUT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: WADI2OUT4 input was triggered 0: WADI2OUT4 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-92. XBARFLG7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	WADI2OUT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: WADI2OUT3 input was triggered 0: WADI2OUT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
26	WADI2OUT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: WADI2OUT2 input was triggered 0: WADI2OUT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
25	WADI2OUT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: WADI2OUT1 input was triggered 0: WADI2OUT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
24	WADI2OUT0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: WADI2OUT0 input was triggered 0: WADI2OUT0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
23	RESERVED	R	0h	Reserved
22	RESERVED	R	0h	Reserved
21	RESERVED	R	0h	Reserved
20	RESERVED	R	0h	Reserved
19	EXTSYNCOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EXTSYNCOUT input was triggered 0: EXTSYNCOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
18	ESMGENEVT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ESMGENEVT input was triggered 0: ESMGENEVT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
17	ADCSOCB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCSOCB input was triggered 0: ADCSOCB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-92. XBARFLG7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	ADCSOCA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCSOCA input was triggered 0: ADCSOCA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	ECAP6_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ECAP6_TRIPOUT input was triggered 0: ECAP6_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
12	ECAP5_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ECAP5_TRIPOUT input was triggered 0: ECAP5_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
11	ECAP4_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ECAP4_TRIPOUT input was triggered 0: ECAP4_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	ECAP3_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ECAP3_TRIPOUT input was triggered 0: ECAP3_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	ECAP2_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ECAP2_TRIPOUT input was triggered 0: ECAP2_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
8	ECAP1_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ECAP1_TRIPOUT input was triggered 0: ECAP1_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved

Table 20-92. XBARFLG7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	ECAP6_OUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ECAP6_OUT input was triggered 0: ECAP6_OUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	ECAP5_OUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ECAP5_OUT input was triggered 0: ECAP5_OUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
3	ECAP4_OUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ECAP4_OUT input was triggered 0: ECAP4_OUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	ECAP3_OUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ECAP3_OUT input was triggered 0: ECAP3_OUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	ECAP2_OUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ECAP2_OUT input was triggered 0: ECAP2_OUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	ECAP1_OUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ECAP1_OUT input was triggered 0: ECAP1_OUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.8 XBARFLG8 Register (Offset = 1Ch) [Reset = 00000000h]

XBARFLG8 is shown in Figure 20-70 and described in Table 20-93.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-70. XBARFLG8 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	EPWM18_TRIPOUT	EPWM17_TRIPOUT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
EPWM16_TRIPOUT	EPWM15_TRIPOUT	EPWM14_TRIPOUT	EPWM13_TRIPOUT	EPWM12_TRIPOUT	EPWM11_TRIPOUT	EPWM10_TRIPOUT	EPWM9_TRIPOUT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
EPWM8_TRIPOUT	EPWM7_TRIPOUT	EPWM6_TRIPOUT	EPWM5_TRIPOUT	EPWM4_TRIPOUT	EPWM3_TRIPOUT	EPWM2_TRIPOUT	EPWM1_TRIPOUT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-93. XBARFLG8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	RESERVED	R	0h	Reserved
29	RESERVED	R	0h	Reserved
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	RESERVED	R	0h	Reserved
24	RESERVED	R	0h	Reserved
23	RESERVED	R	0h	Reserved
22	RESERVED	R	0h	Reserved
21	RESERVED	R	0h	Reserved
20	RESERVED	R	0h	Reserved
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	EPWM18_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM18_TRIPOUT input was triggered 0: EPWM18_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-93. XBARFLG8 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	EPWM17_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM17_TRIPOUT input was triggered 0: EPWM17_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	EPWM16_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM16_TRIPOUT input was triggered 0: EPWM16_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	EPWM15_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM15_TRIPOUT input was triggered 0: EPWM15_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
13	EPWM14_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM14_TRIPOUT input was triggered 0: EPWM14_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
12	EPWM13_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM13_TRIPOUT input was triggered 0: EPWM13_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
11	EPWM12_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM12_TRIPOUT input was triggered 0: EPWM12_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	EPWM11_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM11_TRIPOUT input was triggered 0: EPWM11_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	EPWM10_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM10_TRIPOUT input was triggered 0: EPWM10_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-93. XBARFLG8 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	EPWM9_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM9_TRIPOUT input was triggered 0: EPWM9_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	EPWM8_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM8_TRIPOUT input was triggered 0: EPWM8_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	EPWM7_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM7_TRIPOUT input was triggered 0: EPWM7_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	EPWM6_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM6_TRIPOUT input was triggered 0: EPWM6_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	EPWM5_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM5_TRIPOUT input was triggered 0: EPWM5_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
3	EPWM4_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM4_TRIPOUT input was triggered 0: EPWM4_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	EPWM3_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM3_TRIPOUT input was triggered 0: EPWM3_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	EPWM2_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM2_TRIPOUT input was triggered 0: EPWM2_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-93. XBARFLG8 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	EPWM1_TRIPOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM1_TRIPOUT input was triggered 0: EPWM1_TRIPOUT Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.9 XBARFLG9 Register (Offset = 20h) [Reset = 00000000h]

XBARFLG9 is shown in Figure 20-71 and described in Table 20-94.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-71. XBARFLG9 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	EPWM18_DEL_TRIP	EPWM17_DEL_TRIP
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
EPWM16_DEL_TRIP	EPWM15_DEL_TRIP	EPWM14_DEL_TRIP	EPWM13_DEL_TRIP	EPWM12_DEL_TRIP	EPWM11_DEL_TRIP	EPWM10_DEL_TRIP	EPWM9_DEL_TRIP
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
EPWM8_DEL_TRIP	EPWM7_DEL_TRIP	EPWM6_DEL_TRIP	EPWM5_DEL_TRIP	EPWM4_DEL_TRIP	EPWM3_DEL_TRIP	EPWM2_DEL_TRIP	EPWM1_DEL_TRIP
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-94. XBARFLG9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	RESERVED	R	0h	Reserved
29	RESERVED	R	0h	Reserved
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	RESERVED	R	0h	Reserved
24	RESERVED	R	0h	Reserved
23	RESERVED	R	0h	Reserved
22	RESERVED	R	0h	Reserved
21	RESERVED	R	0h	Reserved
20	RESERVED	R	0h	Reserved
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	EPWM18_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM18_DEL_TRIP input was triggered 0: EPWM18_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-94. XBARFLG9 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	EPWM17_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM17_DEL_TRIP input was triggered 0: EPWM17_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	EPWM16_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM16_DEL_TRIP input was triggered 0: EPWM16_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	EPWM15_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM15_DEL_TRIP input was triggered 0: EPWM15_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
13	EPWM14_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM14_DEL_TRIP input was triggered 0: EPWM14_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
12	EPWM13_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM13_DEL_TRIP input was triggered 0: EPWM13_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
11	EPWM12_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM12_DEL_TRIP input was triggered 0: EPWM12_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	EPWM11_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM11_DEL_TRIP input was triggered 0: EPWM11_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	EPWM10_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM10_DEL_TRIP input was triggered 0: EPWM10_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-94. XBARFLG9 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	EPWM9_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM9_DEL_TRIP input was triggered 0: EPWM9_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	EPWM8_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM8_DEL_TRIP input was triggered 0: EPWM8_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	EPWM7_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM7_DEL_TRIP input was triggered 0: EPWM7_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	EPWM6_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM6_DEL_TRIP input was triggered 0: EPWM6_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	EPWM5_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM5_DEL_TRIP input was triggered 0: EPWM5_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
3	EPWM4_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM4_DEL_TRIP input was triggered 0: EPWM4_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	EPWM3_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM3_DEL_TRIP input was triggered 0: EPWM3_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	EPWM2_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM2_DEL_TRIP input was triggered 0: EPWM2_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-94. XBARFLG9 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	EPWM1_DEL_TRIP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM1_DEL_TRIP input was triggered 0: EPWM1_DEL_TRIP Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.10 XBARFLG10 Register (Offset = 24h) [Reset = 00000000h]

XBARFLG10 is shown in Figure 20-72 and described in Table 20-95.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-72. XBARFLG10 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	EPWM18_DEL_ACTIVE	EPWM17_DEL_ACTIVE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
EPWM16_DEL_ACTIVE	EPWM15_DEL_ACTIVE	EPWM14_DEL_ACTIVE	EPWM13_DEL_ACTIVE	EPWM12_DEL_ACTIVE	EPWM11_DEL_ACTIVE	EPWM10_DEL_ACTIVE	EPWM9_DEL_ACTIVE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
EPWM8_DEL_ACTIVE	EPWM7_DEL_ACTIVE	EPWM6_DEL_ACTIVE	EPWM5_DEL_ACTIVE	EPWM4_DEL_ACTIVE	EPWM3_DEL_ACTIVE	EPWM2_DEL_ACTIVE	EPWM1_DEL_ACTIVE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-95. XBARFLG10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	RESERVED	R	0h	Reserved
29	RESERVED	R	0h	Reserved
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	RESERVED	R	0h	Reserved
24	RESERVED	R	0h	Reserved
23	RESERVED	R	0h	Reserved
22	RESERVED	R	0h	Reserved
21	RESERVED	R	0h	Reserved
20	RESERVED	R	0h	Reserved
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	EPWM18_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM18_DEL_ACTIVE input was triggered 0: EPWM18_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-95. XBARFLG10 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	EPWM17_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM17_DEL_ACTIVE input was triggered 0: EPWM17_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	EPWM16_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM16_DEL_ACTIVE input was triggered 0: EPWM16_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	EPWM15_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM15_DEL_ACTIVE input was triggered 0: EPWM15_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
13	EPWM14_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM14_DEL_ACTIVE input was triggered 0: EPWM14_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
12	EPWM13_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM13_DEL_ACTIVE input was triggered 0: EPWM13_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
11	EPWM12_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM12_DEL_ACTIVE input was triggered 0: EPWM12_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	EPWM11_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM11_DEL_ACTIVE input was triggered 0: EPWM11_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	EPWM10_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM10_DEL_ACTIVE input was triggered 0: EPWM10_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-95. XBARFLG10 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	EPWM9_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM9_DEL_ACTIVE input was triggered 0: EPWM9_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	EPWM8_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM8_DEL_ACTIVE input was triggered 0: EPWM8_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	EPWM7_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM7_DEL_ACTIVE input was triggered 0: EPWM7_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	EPWM6_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM6_DEL_ACTIVE input was triggered 0: EPWM6_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	EPWM5_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM5_DEL_ACTIVE input was triggered 0: EPWM5_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
3	EPWM4_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM4_DEL_ACTIVE input was triggered 0: EPWM4_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	EPWM3_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM3_DEL_ACTIVE input was triggered 0: EPWM3_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	EPWM2_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM2_DEL_ACTIVE input was triggered 0: EPWM2_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-95. XBARFLG10 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	EPWM1_DEL_ACTIVE	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM1_DEL_ACTIVE input was triggered 0: EPWM1_DEL_ACTIVE Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.11 XBARFLG11 Register (Offset = 28h) [Reset = 00000000h]

XBARFLG11 is shown in Figure 20-73 and described in Table 20-96.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-73. XBARFLG11 Register

31	30	29	28	27	26	25	24
EPWM16_B0_sclk	EPWM16_A0_sclk	EPWM15_B0_sclk	EPWM15_A0_sclk	EPWM14_B0_sclk	EPWM14_A0_sclk	EPWM13_B0_sclk	EPWM13_A0_sclk
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
EPWM12_B0_sclk	EPWM12_A0_sclk	EPWM11_B0_sclk	EPWM11_A0_sclk	EPWM10_B0_sclk	EPWM10_A0_sclk	EPWM9_B0_sclk	EPWM9_A0_sclk
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
EPWM8_B0_sclk	EPWM8_A0_sclk	EPWM7_B0_sclk	EPWM7_A0_sclk	EPWM6_B0_sclk	EPWM6_A0_sclk	EPWM5_B0_sclk	EPWM5_A0_sclk
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
EPWM4_B0_sclk	EPWM4_A0_sclk	EPWM3_B0_sclk	EPWM3_A0_sclk	EPWM2_B0_sclk	EPWM2_A0_sclk	EPWM1_B0_sclk	EPWM1_A0_sclk
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-96. XBARFLG11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	EPWM16_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM16_B0_sclk input was triggered 0: EPWM16_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
30	EPWM16_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM16_A0_sclk input was triggered 0: EPWM16_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
29	EPWM15_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM15_B0_sclk input was triggered 0: EPWM15_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-96. XBARFLG11 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	EPWM15_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM15_A0_sclk input was triggered 0: EPWM15_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
27	EPWM14_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM14_B0_sclk input was triggered 0: EPWM14_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
26	EPWM14_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM14_A0_sclk input was triggered 0: EPWM14_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
25	EPWM13_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM13_B0_sclk input was triggered 0: EPWM13_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
24	EPWM13_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM13_A0_sclk input was triggered 0: EPWM13_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
23	EPWM12_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM12_B0_sclk input was triggered 0: EPWM12_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
22	EPWM12_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM12_A0_sclk input was triggered 0: EPWM12_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
21	EPWM11_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM11_B0_sclk input was triggered 0: EPWM11_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-96. XBARFLG11 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	EPWM11_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM11_A0_sclk input was triggered 0: EPWM11_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
19	EPWM10_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM10_B0_sclk input was triggered 0: EPWM10_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
18	EPWM10_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM10_A0_sclk input was triggered 0: EPWM10_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
17	EPWM9_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM9_B0_sclk input was triggered 0: EPWM9_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
16	EPWM9_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM9_A0_sclk input was triggered 0: EPWM9_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	EPWM8_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM8_B0_sclk input was triggered 0: EPWM8_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	EPWM8_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM8_A0_sclk input was triggered 0: EPWM8_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
13	EPWM7_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM7_B0_sclk input was triggered 0: EPWM7_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-96. XBARFLG11 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	EPWM7_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM7_A0_sclk input was triggered 0: EPWM7_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
11	EPWM6_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM6_B0_sclk input was triggered 0: EPWM6_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	EPWM6_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM6_A0_sclk input was triggered 0: EPWM6_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	EPWM5_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM5_B0_sclk input was triggered 0: EPWM5_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
8	EPWM5_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM5_A0_sclk input was triggered 0: EPWM5_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	EPWM4_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM4_B0_sclk input was triggered 0: EPWM4_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	EPWM4_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM4_A0_sclk input was triggered 0: EPWM4_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	EPWM3_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM3_B0_sclk input was triggered 0: EPWM3_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-96. XBARFLG11 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	EPWM3_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM3_A0_sclk input was triggered 0: EPWM3_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
3	EPWM2_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM2_B0_sclk input was triggered 0: EPWM2_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	EPWM2_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM2_A0_sclk input was triggered 0: EPWM2_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	EPWM1_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM1_B0_sclk input was triggered 0: EPWM1_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	EPWM1_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM1_A0_sclk input was triggered 0: EPWM1_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.12 XBARFLG12 Register (Offset = 2Ch) [Reset = 00000000h]

XBARFLG12 is shown in Figure 20-74 and described in Table 20-97.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-74. XBARFLG12 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	EPWM18_B0_sclk	EPWM18_A0_sclk	EPWM17_B0_sclk	EPWM17_A0_sclk
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-97. XBARFLG12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	RESERVED	R	0h	Reserved
29	RESERVED	R	0h	Reserved
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	RESERVED	R	0h	Reserved
24	RESERVED	R	0h	Reserved
23	RESERVED	R	0h	Reserved
22	RESERVED	R	0h	Reserved
21	RESERVED	R	0h	Reserved
20	RESERVED	R	0h	Reserved
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	RESERVED	R	0h	Reserved
16	RESERVED	R	0h	Reserved
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved

Table 20-97. XBARFLG12 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	EPWM18_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM18_B0_sclk input was triggered 0: EPWM18_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	EPWM18_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM18_A0_sclk input was triggered 0: EPWM18_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	EPWM17_B0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM17_B0_sclk input was triggered 0: EPWM17_B0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	EPWM17_A0_sclk	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPWM17_A0_sclk input was triggered 0: EPWM17_A0_sclk Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.13 XBARFLG13 Register (Offset = 30h) [Reset = 00000000h]

XBARFLG13 is shown in Figure 20-75 and described in Table 20-98.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-75. XBARFLG13 Register

31	30	29	28	27	26	25	24
MDL16_OUTB	MDL16_OUTA	MDL15_OUTB	MDL15_OUTA	MDL14_OUTB	MDL14_OUTA	MDL13_OUTB	MDL13_OUTA
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
MDL12_OUTB	MDL12_OUTA	MDL11_OUTB	MDL11_OUTA	MDL10_OUTB	MDL10_OUTA	MDL9_OUTB	MDL9_OUTA
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
MDL8_OUTB	MDL8_OUTA	MDL7_OUTB	MDL7_OUTA	MDL6_OUTB	MDL6_OUTA	MDL5_OUTB	MDL5_OUTA
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
MDL4_OUTB	MDL4_OUTA	MDL3_OUTB	MDL3_OUTA	MDL2_OUTB	MDL2_OUTA	MDL1_OUTB	MDL1_OUTA
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-98. XBARFLG13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MDL16_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL16_OUTB input was triggered 0: MDL16_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
30	MDL16_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL16_OUTA input was triggered 0: MDL16_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
29	MDL15_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL15_OUTB input was triggered 0: MDL15_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
28	MDL15_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL15_OUTA input was triggered 0: MDL15_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-98. XBARFLG13 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	MDL14_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL14_OUTB input was triggered 0: MDL14_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
26	MDL14_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL14_OUTA input was triggered 0: MDL14_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
25	MDL13_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL13_OUTB input was triggered 0: MDL13_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
24	MDL13_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL13_OUTA input was triggered 0: MDL13_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
23	MDL12_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL12_OUTB input was triggered 0: MDL12_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
22	MDL12_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL12_OUTA input was triggered 0: MDL12_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
21	MDL11_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL11_OUTB input was triggered 0: MDL11_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
20	MDL11_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL11_OUTA input was triggered 0: MDL11_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-98. XBARFLG13 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	MDL10_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL10_OUTB input was triggered 0: MDL10_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
18	MDL10_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL10_OUTA input was triggered 0: MDL10_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
17	MDL9_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL9_OUTB input was triggered 0: MDL9_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
16	MDL9_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL9_OUTA input was triggered 0: MDL9_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	MDL8_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL8_OUTB input was triggered 0: MDL8_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	MDL8_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL8_OUTA input was triggered 0: MDL8_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
13	MDL7_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL7_OUTB input was triggered 0: MDL7_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
12	MDL7_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL7_OUTA input was triggered 0: MDL7_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-98. XBARFLG13 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	MDL6_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL6_OUTB input was triggered 0: MDL6_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	MDL6_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL6_OUTA input was triggered 0: MDL6_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	MDL5_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL5_OUTB input was triggered 0: MDL5_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
8	MDL5_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL5_OUTA input was triggered 0: MDL5_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	MDL4_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL4_OUTB input was triggered 0: MDL4_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	MDL4_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL4_OUTA input was triggered 0: MDL4_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	MDL3_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL3_OUTB input was triggered 0: MDL3_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	MDL3_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL3_OUTA input was triggered 0: MDL3_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-98. XBARFLG13 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	MDL2_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL2_OUTB input was triggered 0: MDL2_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	MDL2_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL2_OUTA input was triggered 0: MDL2_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	MDL1_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL1_OUTB input was triggered 0: MDL1_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	MDL1_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL1_OUTA input was triggered 0: MDL1_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.14 XBARFLG14 Register (Offset = 34h) [Reset = 00000000h]

XBARFLG14 is shown in Figure 20-76 and described in Table 20-99.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-76. XBARFLG14 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	MDL18_OUTB	MDL18_OUTA	MDL17_OUTB	MDL17_OUTA
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-99. XBARFLG14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	RESERVED	R	0h	Reserved
29	RESERVED	R	0h	Reserved
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	RESERVED	R	0h	Reserved
24	RESERVED	R	0h	Reserved
23	RESERVED	R	0h	Reserved
22	RESERVED	R	0h	Reserved
21	RESERVED	R	0h	Reserved
20	RESERVED	R	0h	Reserved
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	RESERVED	R	0h	Reserved
16	RESERVED	R	0h	Reserved
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	RESERVED	R	0h	Reserved

Table 20-99. XBARFLG14 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	MDL18_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL18_OUTB input was triggered 0: MDL18_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	MDL18_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL18_OUTA input was triggered 0: MDL18_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	MDL17_OUTB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL17_OUTB input was triggered 0: MDL17_OUTB Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	MDL17_OUTA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MDL17_OUTA input was triggered 0: MDL17_OUTA Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.15 XBARFLG15 Register (Offset = 38h) [Reset = 00000000h]

XBARFLG15 is shown in Figure 20-77 and described in Table 20-100.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-77. XBARFLG15 Register

31	30	29	28	27	26	25	24
CLB6_OUT1	CLB6_OUT0	CLB5_OUT7	CLB5_OUT6	CLB5_OUT3	CLB5_OUT2	CLB5_OUT1	CLB5_OUT0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
CLB4_OUT7	CLB4_OUT6	CLB4_OUT3	CLB4_OUT2	CLB4_OUT1	CLB4_OUT0	CLB3_OUT7	CLB3_OUT6
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
CLB3_OUT3	CLB3_OUT2	CLB3_OUT1	CLB3_OUT0	CLB2_OUT7	CLB2_OUT6	CLB2_OUT3	CLB2_OUT2
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
CLB2_OUT1	CLB2_OUT0	CLB1_OUT7	CLB1_OUT6	CLB1_OUT3	CLB1_OUT2	CLB1_OUT1	CLB1_OUT0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-100. XBARFLG15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CLB6_OUT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB6_OUT1 input was triggered 0: CLB6_OUT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
30	CLB6_OUT0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB6_OUT0 input was triggered 0: CLB6_OUT0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
29	CLB5_OUT7	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB5_OUT7 input was triggered 0: CLB5_OUT7 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
28	CLB5_OUT6	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB5_OUT6 input was triggered 0: CLB5_OUT6 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-100. XBARFLG15 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	CLB5_OUT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB5_OUT3 input was triggered 0: CLB5_OUT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
26	CLB5_OUT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB5_OUT2 input was triggered 0: CLB5_OUT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
25	CLB5_OUT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB5_OUT1 input was triggered 0: CLB5_OUT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
24	CLB5_OUT0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB5_OUT0 input was triggered 0: CLB5_OUT0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
23	CLB4_OUT7	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB4_OUT7 input was triggered 0: CLB4_OUT7 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
22	CLB4_OUT6	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB4_OUT6 input was triggered 0: CLB4_OUT6 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
21	CLB4_OUT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB4_OUT3 input was triggered 0: CLB4_OUT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
20	CLB4_OUT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB4_OUT2 input was triggered 0: CLB4_OUT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-100. XBARFLG15 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	CLB4_OUT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB4_OUT1 input was triggered 0: CLB4_OUT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
18	CLB4_OUT0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB4_OUT0 input was triggered 0: CLB4_OUT0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
17	CLB3_OUT7	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB3_OUT7 input was triggered 0: CLB3_OUT7 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
16	CLB3_OUT6	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB3_OUT6 input was triggered 0: CLB3_OUT6 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	CLB3_OUT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB3_OUT3 input was triggered 0: CLB3_OUT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	CLB3_OUT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB3_OUT2 input was triggered 0: CLB3_OUT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
13	CLB3_OUT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB3_OUT1 input was triggered 0: CLB3_OUT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
12	CLB3_OUT0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB3_OUT0 input was triggered 0: CLB3_OUT0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-100. XBARFLG15 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	CLB2_OUT7	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB2_OUT7 input was triggered 0: CLB2_OUT7 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	CLB2_OUT6	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB2_OUT6 input was triggered 0: CLB2_OUT6 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	CLB2_OUT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB2_OUT3 input was triggered 0: CLB2_OUT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
8	CLB2_OUT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB2_OUT2 input was triggered 0: CLB2_OUT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	CLB2_OUT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB2_OUT1 input was triggered 0: CLB2_OUT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	CLB2_OUT0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB2_OUT0 input was triggered 0: CLB2_OUT0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	CLB1_OUT7	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB1_OUT7 input was triggered 0: CLB1_OUT7 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	CLB1_OUT6	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB1_OUT6 input was triggered 0: CLB1_OUT6 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-100. XBARFLG15 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	CLB1_OUT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB1_OUT3 input was triggered 0: CLB1_OUT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	CLB1_OUT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB1_OUT2 input was triggered 0: CLB1_OUT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	CLB1_OUT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB1_OUT1 input was triggered 0: CLB1_OUT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	CLB1_OUT0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB1_OUT0 input was triggered 0: CLB1_OUT0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.16 XBARFLG16 Register (Offset = 3Ch) [Reset = 00000000h]

XBARFLG16 is shown in Figure 20-78 and described in Table 20-101.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-78. XBARFLG16 Register

31	30	29	28	27	26	25	24
EPG1_EPGOUT3	EPG1_EPGOUT2	EPG1_EPGOUT1	EPG1_EPGOUT0	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
ADCE_EXTMUXSEL3	ADCE_EXTMUXSEL2	ADCE_EXTMUXSEL1	ADCE_EXTMUXSEL0	ADCD_EXTMUXSEL3	ADCD_EXTMUXSEL2	ADCD_EXTMUXSEL1	ADCD_EXTMUXSEL0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
ADCC_EXTMUXSEL3	ADCC_EXTMUXSEL2	ADCC_EXTMUXSEL1	ADCC_EXTMUXSEL0	ADCB_EXTMUXSEL3	ADCB_EXTMUXSEL2	ADCB_EXTMUXSEL1	ADCB_EXTMUXSEL0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
ADCA_EXTMUXSEL3	ADCA_EXTMUXSEL2	ADCA_EXTMUXSEL1	ADCA_EXTMUXSEL0	CLB6_OUT7	CLB6_OUT6	CLB6_OUT3	CLB6_OUT2
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-101. XBARFLG16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	EPG1_EPGOUT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPG1_EPGOUT3 input was triggered 0: EPG1_EPGOUT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
30	EPG1_EPGOUT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPG1_EPGOUT2 input was triggered 0: EPG1_EPGOUT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
29	EPG1_EPGOUT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPG1_EPGOUT1 input was triggered 0: EPG1_EPGOUT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-101. XBARFLG16 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	EPG1_EPGOUT0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: EPG1_EPGOUT0 input was triggered 0: EPG1_EPGOUT0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	RESERVED	R	0h	Reserved
24	RESERVED	R	0h	Reserved
23	ADCE_EXTMUXSEL3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCE_EXTMUXSEL3 input was triggered 0: ADCE_EXTMUXSEL3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
22	ADCE_EXTMUXSEL2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCE_EXTMUXSEL2 input was triggered 0: ADCE_EXTMUXSEL2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
21	ADCE_EXTMUXSEL1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCE_EXTMUXSEL1 input was triggered 0: ADCE_EXTMUXSEL1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
20	ADCE_EXTMUXSEL0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCE_EXTMUXSEL0 input was triggered 0: ADCE_EXTMUXSEL0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
19	ADCD_EXTMUXSEL3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCD_EXTMUXSEL3 input was triggered 0: ADCD_EXTMUXSEL3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
18	ADCD_EXTMUXSEL2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCD_EXTMUXSEL2 input was triggered 0: ADCD_EXTMUXSEL2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-101. XBARFLG16 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	ADCD_EXTMUXSEL1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCD_EXTMUXSEL1 input was triggered 0: ADCD_EXTMUXSEL1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
16	ADCD_EXTMUXSEL0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCD_EXTMUXSEL0 input was triggered 0: ADCD_EXTMUXSEL0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	ADCC_EXTMUXSEL3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCC_EXTMUXSEL3 input was triggered 0: ADCC_EXTMUXSEL3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	ADCC_EXTMUXSEL2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCC_EXTMUXSEL2 input was triggered 0: ADCC_EXTMUXSEL2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
13	ADCC_EXTMUXSEL1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCC_EXTMUXSEL1 input was triggered 0: ADCC_EXTMUXSEL1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
12	ADCC_EXTMUXSEL0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCC_EXTMUXSEL0 input was triggered 0: ADCC_EXTMUXSEL0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
11	ADCB_EXTMUXSEL3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCB_EXTMUXSEL3 input was triggered 0: ADCB_EXTMUXSEL3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	ADCB_EXTMUXSEL2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCB_EXTMUXSEL2 input was triggered 0: ADCB_EXTMUXSEL2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-101. XBARFLG16 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	ADCB_EXTMUXSEL1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCB_EXTMUXSEL1 input was triggered 0: ADCB_EXTMUXSEL1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
8	ADCB_EXTMUXSEL0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCB_EXTMUXSEL0 input was triggered 0: ADCB_EXTMUXSEL0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	ADCA_EXTMUXSEL3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCA_EXTMUXSEL3 input was triggered 0: ADCA_EXTMUXSEL3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	ADCA_EXTMUXSEL2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCA_EXTMUXSEL2 input was triggered 0: ADCA_EXTMUXSEL2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	ADCA_EXTMUXSEL1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCA_EXTMUXSEL1 input was triggered 0: ADCA_EXTMUXSEL1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	ADCA_EXTMUXSEL0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ADCA_EXTMUXSEL0 input was triggered 0: ADCA_EXTMUXSEL0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
3	CLB6_OUT7	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB6_OUT7 input was triggered 0: CLB6_OUT7 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	CLB6_OUT6	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB6_OUT6 input was triggered 0: CLB6_OUT6 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-101. XBARFLG16 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	CLB6_OUT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB6_OUT3 input was triggered 0: CLB6_OUT3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	CLB6_OUT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CLB6_OUT2 input was triggered 0: CLB6_OUT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.17 XBARFLG17 Register (Offset = 40h) [Reset = 00000000h]

XBARFLG17 is shown in Figure 20-79 and described in Table 20-102.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-79. XBARFLG17 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						MCANF_FEVT2	MCANF_FEVT1
R-0h						R-0h	R-0h
15	14	13	12	11	10	9	8
MCANF_FEVT0	MCANE_FEVT2	MCANE_FEVT1	MCANE_FEVT0	MCAND_FEVT2	MCAND_FEVT1	MCAND_FEVT0	MCANC_FEVT2
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
MCANC_FEVT1	MCANC_FEVT0	MCANB_FEVT2	MCANB_FEVT1	MCANB_FEVT0	MCANA_FEVT2	MCANA_FEVT1	MCANA_FEVT0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-102. XBARFLG17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17	MCANF_FEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCANF_FEVT2 input was triggered 0: MCANF_FEVT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
16	MCANF_FEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCANF_FEVT1 input was triggered 0: MCANF_FEVT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	MCANF_FEVT0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCANF_FEVT0 input was triggered 0: MCANF_FEVT0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-102. XBARFLG17 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	MCANE_FEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCANE_FEVT2 input was triggered 0: MCANE_FEVT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
13	MCANE_FEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCANE_FEVT1 input was triggered 0: MCANE_FEVT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
12	MCANE_FEVT0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCANE_FEVT0 input was triggered 0: MCANE_FEVT0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
11	MCAND_FEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCAND_FEVT2 input was triggered 0: MCAND_FEVT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	MCAND_FEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCAND_FEVT1 input was triggered 0: MCAND_FEVT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	MCAND_FEVT0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCAND_FEVT0 input was triggered 0: MCAND_FEVT0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
8	MCANC_FEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCANC_FEVT2 input was triggered 0: MCANC_FEVT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	MCANC_FEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCANC_FEVT1 input was triggered 0: MCANC_FEVT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-102. XBARFLG17 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	MCANC_FEVT0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCANC_FEVT0 input was triggered 0: MCANC_FEVT0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	MCANB_FEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCANB_FEVT2 input was triggered 0: MCANB_FEVT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	MCANB_FEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCANB_FEVT1 input was triggered 0: MCANB_FEVT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
3	MCANB_FEVT0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCANB_FEVT0 input was triggered 0: MCANB_FEVT0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	MCANA_FEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCANA_FEVT2 input was triggered 0: MCANA_FEVT2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	MCANA_FEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCANA_FEVT1 input was triggered 0: MCANA_FEVT1 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	MCANA_FEVT0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: MCANA_FEVT0 input was triggered 0: MCANA_FEVT0 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.18 XBARFLG18 Register (Offset = 44h) [Reset = 00000000h]

XBARFLG18 is shown in Figure 20-80 and described in Table 20-103.

Return to the Summary Table.

This register is used to flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered.
1: Corresponding Input was triggered
0: Corresponding Input was not triggered

Figure 20-80. XBARFLG18 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				CPU3ERADEVT11	CPU3ERADEVT10	CPU3ERADEVT9	CPU3ERADEVT8
R-0h				R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
CPU2ERADEVT11	CPU2ERADEVT10	CPU2ERADEVT9	CPU2ERADEVT8	CPU1ERADEVT11	CPU1ERADEVT10	CPU1ERADEVT9	CPU1ERADEVT8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
FSIRXD_TRIG_3	FSIRXD_TRIG_2	FSIRXC_TRIG_3	FSIRXC_TRIG_2	FSIRXB_TRIG_3	FSIRXB_TRIG_2	FSIRXA_TRIG_3	FSIRXA_TRIG_2
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-103. XBARFLG18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	CPU3ERADEVT11	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU3ERADEVT11 input was triggered 0: CPU3ERADEVT11 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
18	CPU3ERADEVT10	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU3ERADEVT10 input was triggered 0: CPU3ERADEVT10 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
17	CPU3ERADEVT9	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU3ERADEVT9 input was triggered 0: CPU3ERADEVT9 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-103. XBARFLG18 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CPU3ERADEVT8	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU3ERADEVT8 input was triggered 0: CPU3ERADEVT8 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	CPU2ERADEVT11	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU2ERADEVT11 input was triggered 0: CPU2ERADEVT11 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	CPU2ERADEVT10	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU2ERADEVT10 input was triggered 0: CPU2ERADEVT10 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
13	CPU2ERADEVT9	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU2ERADEVT9 input was triggered 0: CPU2ERADEVT9 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
12	CPU2ERADEVT8	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU2ERADEVT8 input was triggered 0: CPU2ERADEVT8 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
11	CPU1ERADEVT11	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU1ERADEVT11 input was triggered 0: CPU1ERADEVT11 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	CPU1ERADEVT10	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU1ERADEVT10 input was triggered 0: CPU1ERADEVT10 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	CPU1ERADEVT9	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU1ERADEVT9 input was triggered 0: CPU1ERADEVT9 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-103. XBARFLG18 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	CPU1ERADEVT8	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU1ERADEVT8 input was triggered 0: CPU1ERADEVT8 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	FSIRXD_TRIG_3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIRXD_TRIG_3 input was triggered 0: FSIRXD_TRIG_3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	FSIRXD_TRIG_2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIRXD_TRIG_2 input was triggered 0: FSIRXD_TRIG_2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	FSIRXC_TRIG_3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIRXC_TRIG_3 input was triggered 0: FSIRXC_TRIG_3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	FSIRXC_TRIG_2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIRXC_TRIG_2 input was triggered 0: FSIRXC_TRIG_2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
3	FSIRXB_TRIG_3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIRXB_TRIG_3 input was triggered 0: FSIRXB_TRIG_3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	FSIRXB_TRIG_2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIRXB_TRIG_2 input was triggered 0: FSIRXB_TRIG_2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	FSIRXA_TRIG_3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIRXA_TRIG_3 input was triggered 0: FSIRXA_TRIG_3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-103. XBARFLG18 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	FSIRXA_TRIG_2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIRXA_TRIG_2 input was triggered 0: FSIRXA_TRIG_2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

20.5.9.19 XBARCLR1 Register (Offset = 100h) [Reset = 00000000h]

XBARCLR1 is shown in Figure 20-81 and described in Table 20-104.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-81. XBARCLR1 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0-0h	R-0-0h	R-0-0h	R-0-0h	R-0-0h	R-0-0h	R-0-0h	R-0-0h
23	22	21	20	19	18	17	16
CMPSS12_CTRIPL	CMPSS12_CTRIPH	CMPSS11_CTRIPL	CMPSS11_CTRIPH	CMPSS10_CTRIPL	CMPSS10_CTRIPH	CMPSS9_CTRIPL	CMPSS9_CTRIPH
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
CMPSS8_CTRIPH	CMPSS8_CTRIPL	CMPSS7_CTRIPH	CMPSS7_CTRIPL	CMPSS6_CTRIPH	CMPSS6_CTRIPL	CMPSS5_CTRIPH	CMPSS5_CTRIPL
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
CMPSS4_CTRIPH	CMPSS4_CTRIPL	CMPSS3_CTRIPH	CMPSS3_CTRIPL	CMPSS2_CTRIPH	CMPSS2_CTRIPL	CMPSS1_CTRIPH	CMPSS1_CTRIPL
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-104. XBARCLR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0	0h	Reserved
30	RESERVED	R-0	0h	Reserved
29	RESERVED	R-0	0h	Reserved
28	RESERVED	R-0	0h	Reserved
27	RESERVED	R-0	0h	Reserved
26	RESERVED	R-0	0h	Reserved
25	RESERVED	R-0	0h	Reserved
24	RESERVED	R-0	0h	Reserved
23	CMPSS12_CTRIPL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS12_CTRIPL bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
22	CMPSS12_CTRIPH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS12_CTRIPH bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
21	CMPSS11_CTRIPL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS11_CTRIPL bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
20	CMPSS11_CTRIPH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS11_CTRIPH bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-104. XBARCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	CMPSS10_CTRIPL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS10_CTRIPL bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
18	CMPSS10_CTRIPH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS10_CTRIPH bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
17	CMPSS9_CTRIPL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS9_CTRIPL bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	CMPSS9_CTRIPH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS9_CTRIPH bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	CMPSS8_CTRIPH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS8_CTRIPH bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
14	CMPSS8_CTRIPL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS8_CTRIPL bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
13	CMPSS7_CTRIPH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS7_CTRIPH bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	CMPSS7_CTRIPL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS7_CTRIPL bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
11	CMPSS6_CTRIPH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS6_CTRIPH bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	CMPSS6_CTRIPL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS6_CTRIPL bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	CMPSS5_CTRIPH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS5_CTRIPH bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	CMPSS5_CTRIPL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS5_CTRIPL bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	CMPSS4_CTRIPH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS4_CTRIPH bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
6	CMPSS4_CTRIPL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS4_CTRIPL bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-104. XBARCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	CMPSS3_CTRIPH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS3_CTRIPH bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	CMPSS3_CTRIPL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS3_CTRIPL bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	CMPSS2_CTRIPH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS2_CTRIPH bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	CMPSS2_CTRIPL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS2_CTRIPL bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	CMPSS1_CTRIPH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS1_CTRIPH bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	CMPSS1_CTRIPL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS1_CTRIPL bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.20 XBARCLR2 Register (Offset = 104h) [Reset = 00000000h]

XBARCLR2 is shown in Figure 20-82 and described in Table 20-105.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-82. XBARCLR2 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0-0h	R-0-0h	R-0-0h	R-0-0h	R-0-0h	R-0-0h	R-0-0h	R-0-0h
23	22	21	20	19	18	17	16
CMPSS12_CTRIPOUTL	CMPSS12_CTRIPOUTH	CMPSS11_CTRIPOUTL	CMPSS11_CTRIPOUTH	CMPSS10_CTRIPOUTL	CMPSS10_CTRIPOUTH	CMPSS9_CTRIPOUTL	CMPSS9_CTRIPOUTH
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
CMPSS8_CTRIPOUTH	CMPSS8_CTRIPOUTL	CMPSS7_CTRIPOUTH	CMPSS7_CTRIPOUTL	CMPSS6_CTRIPOUTH	CMPSS6_CTRIPOUTL	CMPSS5_CTRIPOUTH	CMPSS5_CTRIPOUTL
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
CMPSS4_CTRIPOUTH	CMPSS4_CTRIPOUTL	CMPSS3_CTRIPOUTH	CMPSS3_CTRIPOUTL	CMPSS2_CTRIPOUTH	CMPSS2_CTRIPOUTL	CMPSS1_CTRIPOUTH	CMPSS1_CTRIPOUTL
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-105. XBARCLR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0	0h	Reserved
30	RESERVED	R-0	0h	Reserved
29	RESERVED	R-0	0h	Reserved
28	RESERVED	R-0	0h	Reserved
27	RESERVED	R-0	0h	Reserved
26	RESERVED	R-0	0h	Reserved
25	RESERVED	R-0	0h	Reserved
24	RESERVED	R-0	0h	Reserved
23	CMPSS12_CTRIPOUTL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS12_CTRIPOUTL bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
22	CMPSS12_CTRIPOUTH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS12_CTRIPOUTH bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
21	CMPSS11_CTRIPOUTL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS11_CTRIPOUTL bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
20	CMPSS11_CTRIPOUTH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS11_CTRIPOUTH bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-105. XBARCLR2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	CMPSS10_CTRIPOUTL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS10_CTRIPOUTL bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
18	CMPSS10_CTRIPOUTH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS10_CTRIPOUTH bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
17	CMPSS9_CTRIPOUTL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS9_CTRIPOUTL bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	CMPSS9_CTRIPOUTH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS9_CTRIPOUTH bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	CMPSS8_CTRIPOUTH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS8_CTRIPOUTH bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
14	CMPSS8_CTRIPOUTL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS8_CTRIPOUTL bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
13	CMPSS7_CTRIPOUTH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS7_CTRIPOUTH bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	CMPSS7_CTRIPOUTL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS7_CTRIPOUTL bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
11	CMPSS6_CTRIPOUTH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS6_CTRIPOUTH bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	CMPSS6_CTRIPOUTL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS6_CTRIPOUTL bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	CMPSS5_CTRIPOUTH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS5_CTRIPOUTH bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	CMPSS5_CTRIPOUTL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS5_CTRIPOUTL bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	CMPSS4_CTRIPOUTH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS4_CTRIPOUTH bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
6	CMPSS4_CTRIPOUTL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS4_CTRIPOUTL bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-105. XBARCLR2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	CMPSS3_CTRIPOUTH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS3_CTRIPOUTH bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	CMPSS3_CTRIPOUTL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS3_CTRIPOUTL bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	CMPSS2_CTRIPOUTH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS2_CTRIPOUTH bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	CMPSS2_CTRIPOUTL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS2_CTRIPOUTL bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	CMPSS1_CTRIPOUTH	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS1_CTRIPOUTH bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	CMPSS1_CTRIPOUTL	R-0/W1S	0h	Writing 1 to this bit clears the CMPSS1_CTRIPOUTL bit in XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.21 XBARCLR3 Register (Offset = 108h) [Reset = 00000000h]

XBARCLR3 is shown in Figure 20-83 and described in Table 20-106.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-83. XBARCLR3 Register

31	30	29	28	27	26	25	24
SD4FLT4_COMPL	SD4FLT4_COMPH	SD4FLT3_COMPL	SD4FLT3_COMPH	SD4FLT2_COMPL	SD4FLT2_COMPH	SD4FLT1_COMPL	SD4FLT1_COMPH
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
SD3FLT4_COMPL	SD3FLT4_COMPH	SD3FLT3_COMPL	SD3FLT3_COMPH	SD3FLT2_COMPL	SD3FLT2_COMPH	SD3FLT1_COMPL	SD3FLT1_COMPH
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
SD2FLT4_COMPH	SD2FLT4_COMPL	SD2FLT3_COMPH	SD2FLT3_COMPL	SD2FLT2_COMPH	SD2FLT2_COMPL	SD2FLT1_COMPH	SD2FLT1_COMPL
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
SD1FLT4_COMPH	SD1FLT4_COMPL	SD1FLT3_COMPH	SD1FLT3_COMPL	SD1FLT2_COMPH	SD1FLT2_COMPL	SD1FLT1_COMPH	SD1FLT1_COMPL
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-106. XBARCLR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SD4FLT4_COMPL	R-0/W1S	0h	Writing 1 to this bit clears the SD4FLT4_COMPL bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
30	SD4FLT4_COMPH	R-0/W1S	0h	Writing 1 to this bit clears the SD4FLT4_COMPH bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
29	SD4FLT3_COMPL	R-0/W1S	0h	Writing 1 to this bit clears the SD4FLT3_COMPL bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
28	SD4FLT3_COMPH	R-0/W1S	0h	Writing 1 to this bit clears the SD4FLT3_COMPH bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
27	SD4FLT2_COMPL	R-0/W1S	0h	Writing 1 to this bit clears the SD4FLT2_COMPL bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
26	SD4FLT2_COMPH	R-0/W1S	0h	Writing 1 to this bit clears the SD4FLT2_COMPH bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-106. XBARCLR3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	SD4FLT1_COMPL	R-0/W1S	0h	Writing 1 to this bit clears the SD4FLT1_COMPL bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
24	SD4FLT1_COMPH	R-0/W1S	0h	Writing 1 to this bit clears the SD4FLT1_COMPH bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
23	SD3FLT4_COMPL	R-0/W1S	0h	Writing 1 to this bit clears the SD3FLT4_COMPL bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
22	SD3FLT4_COMPH	R-0/W1S	0h	Writing 1 to this bit clears the SD3FLT4_COMPH bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
21	SD3FLT3_COMPL	R-0/W1S	0h	Writing 1 to this bit clears the SD3FLT3_COMPL bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
20	SD3FLT3_COMPH	R-0/W1S	0h	Writing 1 to this bit clears the SD3FLT3_COMPH bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
19	SD3FLT2_COMPL	R-0/W1S	0h	Writing 1 to this bit clears the SD3FLT2_COMPL bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
18	SD3FLT2_COMPH	R-0/W1S	0h	Writing 1 to this bit clears the SD3FLT2_COMPH bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
17	SD3FLT1_COMPL	R-0/W1S	0h	Writing 1 to this bit clears the SD3FLT1_COMPL bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	SD3FLT1_COMPH	R-0/W1S	0h	Writing 1 to this bit clears the SD3FLT1_COMPH bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	SD2FLT4_COMPH	R-0/W1S	0h	Writing 1 to this bit clears the SD2FLT4_COMPH bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
14	SD2FLT4_COMPL	R-0/W1S	0h	Writing 1 to this bit clears the SD2FLT4_COMPL bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
13	SD2FLT3_COMPH	R-0/W1S	0h	Writing 1 to this bit clears the SD2FLT3_COMPH bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	SD2FLT3_COMPL	R-0/W1S	0h	Writing 1 to this bit clears the SD2FLT3_COMPL bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-106. XBARCLR3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SD2FLT2_COMPH	R-0/W1S	0h	Writing 1 to this bit clears the SD2FLT2_COMPH bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	SD2FLT2_COMPL	R-0/W1S	0h	Writing 1 to this bit clears the SD2FLT2_COMPL bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	SD2FLT1_COMPH	R-0/W1S	0h	Writing 1 to this bit clears the SD2FLT1_COMPH bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	SD2FLT1_COMPL	R-0/W1S	0h	Writing 1 to this bit clears the SD2FLT1_COMPL bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	SD1FLT4_COMPH	R-0/W1S	0h	Writing 1 to this bit clears the SD1FLT4_COMPH bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
6	SD1FLT4_COMPL	R-0/W1S	0h	Writing 1 to this bit clears the SD1FLT4_COMPL bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
5	SD1FLT3_COMPH	R-0/W1S	0h	Writing 1 to this bit clears the SD1FLT3_COMPH bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	SD1FLT3_COMPL	R-0/W1S	0h	Writing 1 to this bit clears the SD1FLT3_COMPL bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	SD1FLT2_COMPH	R-0/W1S	0h	Writing 1 to this bit clears the SD1FLT2_COMPH bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	SD1FLT2_COMPL	R-0/W1S	0h	Writing 1 to this bit clears the SD1FLT2_COMPL bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	SD1FLT1_COMPH	R-0/W1S	0h	Writing 1 to this bit clears the SD1FLT1_COMPH bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	SD1FLT1_COMPL	R-0/W1S	0h	Writing 1 to this bit clears the SD1FLT1_COMPL bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.22 XBARCLR4 Register (Offset = 10Ch) [Reset = 00000000h]

XBARCLR4 is shown in Figure 20-84 and described in Table 20-107.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-84. XBARCLR4 Register

31	30	29	28	27	26	25	24
INPUT32	INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
INPUT24	INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
INPUT16	INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
INPUT8	INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-107. XBARCLR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT32	R-0/W1S	0h	Writing 1 to this bit clears the INPUT32 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
30	INPUT31	R-0/W1S	0h	Writing 1 to this bit clears the INPUT31 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
29	INPUT30	R-0/W1S	0h	Writing 1 to this bit clears the INPUT30 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
28	INPUT29	R-0/W1S	0h	Writing 1 to this bit clears the INPUT29 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
27	INPUT28	R-0/W1S	0h	Writing 1 to this bit clears the INPUT28 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
26	INPUT27	R-0/W1S	0h	Writing 1 to this bit clears the INPUT27 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
25	INPUT26	R-0/W1S	0h	Writing 1 to this bit clears the INPUT26 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
24	INPUT25	R-0/W1S	0h	Writing 1 to this bit clears the INPUT25 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
23	INPUT24	R-0/W1S	0h	Writing 1 to this bit clears the INPUT24 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-107. XBARCLR4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	INPUT23	R-0/W1S	0h	Writing 1 to this bit clears the INPUT23 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
21	INPUT22	R-0/W1S	0h	Writing 1 to this bit clears the INPUT22 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
20	INPUT21	R-0/W1S	0h	Writing 1 to this bit clears the INPUT21 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
19	INPUT20	R-0/W1S	0h	Writing 1 to this bit clears the INPUT20 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
18	INPUT19	R-0/W1S	0h	Writing 1 to this bit clears the INPUT19 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
17	INPUT18	R-0/W1S	0h	Writing 1 to this bit clears the INPUT18 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	INPUT17	R-0/W1S	0h	Writing 1 to this bit clears the INPUT17 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	INPUT16	R-0/W1S	0h	Writing 1 to this bit clears the INPUT16 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
14	INPUT15	R-0/W1S	0h	Writing 1 to this bit clears the INPUT15 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
13	INPUT14	R-0/W1S	0h	Writing 1 to this bit clears the INPUT14 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	INPUT13	R-0/W1S	0h	Writing 1 to this bit clears the INPUT13 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
11	INPUT12	R-0/W1S	0h	Writing 1 to this bit clears the INPUT12 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	INPUT11	R-0/W1S	0h	Writing 1 to this bit clears the INPUT11 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	INPUT10	R-0/W1S	0h	Writing 1 to this bit clears the INPUT10 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	INPUT9	R-0/W1S	0h	Writing 1 to this bit clears the INPUT9 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	INPUT8	R-0/W1S	0h	Writing 1 to this bit clears the INPUT8 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
6	INPUT7	R-0/W1S	0h	Writing 1 to this bit clears the INPUT7 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-107. XBARCLR4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	INPUT6	R-0/W1S	0h	Writing 1 to this bit clears the INPUT6 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	INPUT5	R-0/W1S	0h	Writing 1 to this bit clears the INPUT5 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	INPUT4	R-0/W1S	0h	Writing 1 to this bit clears the INPUT4 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	INPUT3	R-0/W1S	0h	Writing 1 to this bit clears the INPUT3 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	INPUT2	R-0/W1S	0h	Writing 1 to this bit clears the INPUT2 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	INPUT1	R-0/W1S	0h	Writing 1 to this bit clears the INPUT1 bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.23 XBARCLR5 Register (Offset = 110h) [Reset = 00000000h]

XBARCLR5 is shown in Figure 20-85 and described in Table 20-108.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-85. XBARCLR5 Register

31	30	29	28	27	26	25	24
CPU3_ADCCHECKEVT4	CPU3_ADCCHECKEVT3	CPU3_ADCCHECKEVT2	CPU3_ADCCHECKEVT1	CPU2_ADCCHECKEVT4	CPU2_ADCCHECKEVT3	CPU2_ADCCHECKEVT2	CPU2_ADCCHECKEVT1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
CPU1_ADCCHECKEVT4	CPU1_ADCCHECKEVT3	CPU1_ADCCHECKEVT2	CPU1_ADCCHECKEVT1	ADCEEVT4	ADCEEVT3	ADCEEVT2	ADCEEVT1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
ADCDEVT4	ADCDEVT3	ADCDEVT2	ADCDEVT1	ADCCEVT4	ADCCEVT3	ADCCEVT2	ADCCEVT1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
ADCBEVT4	ADCBEVT3	ADCBEVT2	ADCBEVT1	ADCAEVT4	ADCAEVT3	ADCAEVT2	ADCAEVT1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-108. XBARCLR5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CPU3_ADCCHECKEVT4	R-0/W1S	0h	Writing 1 to this bit clears the CPU3_ADCCHECKEVT4 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
30	CPU3_ADCCHECKEVT3	R-0/W1S	0h	Writing 1 to this bit clears the CPU3_ADCCHECKEVT3 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
29	CPU3_ADCCHECKEVT2	R-0/W1S	0h	Writing 1 to this bit clears the CPU3_ADCCHECKEVT2 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
28	CPU3_ADCCHECKEVT1	R-0/W1S	0h	Writing 1 to this bit clears the CPU3_ADCCHECKEVT1 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
27	CPU2_ADCCHECKEVT4	R-0/W1S	0h	Writing 1 to this bit clears the CPU2_ADCCHECKEVT4 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
26	CPU2_ADCCHECKEVT3	R-0/W1S	0h	Writing 1 to this bit clears the CPU2_ADCCHECKEVT3 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-108. XBARCLR5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	CPU2_ADCCHECKEVT2	R-0/W1S	0h	Writing 1 to this bit clears the CPU2_ADCCHECKEVT2 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
24	CPU2_ADCCHECKEVT1	R-0/W1S	0h	Writing 1 to this bit clears the CPU2_ADCCHECKEVT1 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
23	CPU1_ADCCHECKEVT4	R-0/W1S	0h	Writing 1 to this bit clears the CPU1_ADCCHECKEVT4 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
22	CPU1_ADCCHECKEVT3	R-0/W1S	0h	Writing 1 to this bit clears the CPU1_ADCCHECKEVT3 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
21	CPU1_ADCCHECKEVT2	R-0/W1S	0h	Writing 1 to this bit clears the CPU1_ADCCHECKEVT2 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
20	CPU1_ADCCHECKEVT1	R-0/W1S	0h	Writing 1 to this bit clears the CPU1_ADCCHECKEVT1 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
19	ADCEEVT4	R-0/W1S	0h	Writing 1 to this bit clears the ADCEEVT4 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
18	ADCEEVT3	R-0/W1S	0h	Writing 1 to this bit clears the ADCEEVT3 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
17	ADCEEVT2	R-0/W1S	0h	Writing 1 to this bit clears the ADCEEVT2 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	ADCEEVT1	R-0/W1S	0h	Writing 1 to this bit clears the ADCEEVT1 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	ADCDEVT4	R-0/W1S	0h	Writing 1 to this bit clears the ADCDEVT4 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
14	ADCDEVT3	R-0/W1S	0h	Writing 1 to this bit clears the ADCDEVT3 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
13	ADCDEVT2	R-0/W1S	0h	Writing 1 to this bit clears the ADCDEVT2 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	ADCDEVT1	R-0/W1S	0h	Writing 1 to this bit clears the ADCDEVT1 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-108. XBARCLR5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	ADCCEVT4	R-0/W1S	0h	Writing 1 to this bit clears the ADCCEVT4 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	ADCCEVT3	R-0/W1S	0h	Writing 1 to this bit clears the ADCCEVT3 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	ADCCEVT2	R-0/W1S	0h	Writing 1 to this bit clears the ADCCEVT2 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	ADCCEVT1	R-0/W1S	0h	Writing 1 to this bit clears the ADCCEVT1 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	ADCBEVT4	R-0/W1S	0h	Writing 1 to this bit clears the ADCBEVT4 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
6	ADCBEVT3	R-0/W1S	0h	Writing 1 to this bit clears the ADCBEVT3 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
5	ADCBEVT2	R-0/W1S	0h	Writing 1 to this bit clears the ADCBEVT2 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	ADCBEVT1	R-0/W1S	0h	Writing 1 to this bit clears the ADCBEVT1 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	ADCAEVT4	R-0/W1S	0h	Writing 1 to this bit clears the ADCAEVT4 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	ADCAEVT3	R-0/W1S	0h	Writing 1 to this bit clears the ADCAEVT3 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	ADCAEVT2	R-0/W1S	0h	Writing 1 to this bit clears the ADCAEVT2 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	ADCAEVT1	R-0/W1S	0h	Writing 1 to this bit clears the ADCAEVT1 bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.24 XBARCLR6 Register (Offset = 114h) [Reset = 00000000h]

XBARCLR6 is shown in Figure 20-86 and described in Table 20-109.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-86. XBARCLR6 Register

31	30	29	28	27	26	25	24
ECATSYNC1	ECATSYNC0	WADI1OUT7	WADI1OUT6	WADI1OUT5	WADI1OUT4	WADI1OUT3	WADI1OUT2
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
WADI1OUT1	WADI1OUT0	RESERVED	RESERVED	FSID_RX_TRIG 1	FSIC_RX_TRIG 1	FSIB_RX_TRIG 1	FSIA_RX_TRIG 1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	CLB6_OUT5	CLB6_OUT4	CLB5_OUT5	CLB5_OUT4
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
CLB4_OUT5	CLB4_OUT4	CLB3_OUT5	CLB3_OUT4	CLB2_OUT5	CLB2_OUT4	CLB1_OUT5	CLB1_OUT4
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-109. XBARCLR6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ECATSYNC1	R-0/W1S	0h	Writing 1 to this bit clears the ECATSYNC1 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
30	ECATSYNC0	R-0/W1S	0h	Writing 1 to this bit clears the ECATSYNC0 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
29	WADI1OUT7	R-0/W1S	0h	Writing 1 to this bit clears the WADI1OUT7 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
28	WADI1OUT6	R-0/W1S	0h	Writing 1 to this bit clears the WADI1OUT6 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
27	WADI1OUT5	R-0/W1S	0h	Writing 1 to this bit clears the WADI1OUT5 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
26	WADI1OUT4	R-0/W1S	0h	Writing 1 to this bit clears the WADI1OUT4 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
25	WADI1OUT3	R-0/W1S	0h	Writing 1 to this bit clears the WADI1OUT3 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-109. XBARCLR6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	WADI1OUT2	R-0/W1S	0h	Writing 1 to this bit clears the WADI1OUT2 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
23	WADI1OUT1	R-0/W1S	0h	Writing 1 to this bit clears the WADI1OUT1 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
22	WADI1OUT0	R-0/W1S	0h	Writing 1 to this bit clears the WADI1OUT0 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
21	RESERVED	R-0/W1S	0h	Reserved
20	RESERVED	R-0/W1S	0h	Reserved
19	FSID_RX_TRIG1	R-0/W1S	0h	Writing 1 to this bit clears the FSID_RX_TRIG1 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
18	FSIC_RX_TRIG1	R-0/W1S	0h	Writing 1 to this bit clears the FSIC_RX_TRIG1 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
17	FSIB_RX_TRIG1	R-0/W1S	0h	Writing 1 to this bit clears the FSIB_RX_TRIG1 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	FSIA_RX_TRIG1	R-0/W1S	0h	Writing 1 to this bit clears the FSIA_RX_TRIG1 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	RESERVED	R-0/W1S	0h	Reserved
14	RESERVED	R-0/W1S	0h	Reserved
13	RESERVED	R-0/W1S	0h	Reserved
12	RESERVED	R-0/W1S	0h	Reserved
11	CLB6_OUT5	R-0/W1S	0h	Writing 1 to this bit clears the CLB6_OUT5 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	CLB6_OUT4	R-0/W1S	0h	Writing 1 to this bit clears the CLB6_OUT4 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	CLB5_OUT5	R-0/W1S	0h	Writing 1 to this bit clears the CLB5_OUT5 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	CLB5_OUT4	R-0/W1S	0h	Writing 1 to this bit clears the CLB5_OUT4 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	CLB4_OUT5	R-0/W1S	0h	Writing 1 to this bit clears the CLB4_OUT5 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-109. XBARCLR6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	CLB4_OUT4	R-0/W1S	0h	Writing 1 to this bit clears the CLB4_OUT4 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
5	CLB3_OUT5	R-0/W1S	0h	Writing 1 to this bit clears the CLB3_OUT5 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	CLB3_OUT4	R-0/W1S	0h	Writing 1 to this bit clears the CLB3_OUT4 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	CLB2_OUT5	R-0/W1S	0h	Writing 1 to this bit clears the CLB2_OUT5 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	CLB2_OUT4	R-0/W1S	0h	Writing 1 to this bit clears the CLB2_OUT4 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	CLB1_OUT5	R-0/W1S	0h	Writing 1 to this bit clears the CLB1_OUT5 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	CLB1_OUT4	R-0/W1S	0h	Writing 1 to this bit clears the CLB1_OUT4 bit in the XBARFLG6 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.25 XBARCLR7 Register (Offset = 118h) [Reset = 00000000h]

XBARCLR7 is shown in Figure 20-87 and described in Table 20-110.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-87. XBARCLR7 Register

31	30	29	28	27	26	25	24
WADI2OUT7	WADI2OUT6	WADI2OUT5	WADI2OUT4	WADI2OUT3	WADI2OUT2	WADI2OUT1	WADI2OUT0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	EXTSYNCOUT	ESMGENEVT	ADCSOCB	ADCSOCA
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	ECAP6_TRIPOUT	ECAP5_TRIPOUT	ECAP4_TRIPOUT	ECAP3_TRIPOUT	ECAP2_TRIPOUT	ECAP1_TRIPOUT
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	ECAP6_OUT	ECAP5_OUT	ECAP4_OUT	ECAP3_OUT	ECAP2_OUT	ECAP1_OUT
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-110. XBARCLR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	WADI2OUT7	R-0/W1S	0h	Writing 1 to this bit clears the WADI2OUT7 bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
30	WADI2OUT6	R-0/W1S	0h	Writing 1 to this bit clears the WADI2OUT6 bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
29	WADI2OUT5	R-0/W1S	0h	Writing 1 to this bit clears the WADI2OUT5 bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
28	WADI2OUT4	R-0/W1S	0h	Writing 1 to this bit clears the WADI2OUT4 bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
27	WADI2OUT3	R-0/W1S	0h	Writing 1 to this bit clears the WADI2OUT3 bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
26	WADI2OUT2	R-0/W1S	0h	Writing 1 to this bit clears the WADI2OUT2 bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
25	WADI2OUT1	R-0/W1S	0h	Writing 1 to this bit clears the WADI2OUT1 bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-110. XBARCLR7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	WADI2OUT0	R-0/W1S	0h	Writing 1 to this bit clears the WADI2OUT0 bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
23	RESERVED	R-0/W1S	0h	Reserved
22	RESERVED	R-0/W1S	0h	Reserved
21	RESERVED	R-0/W1S	0h	Reserved
20	RESERVED	R-0/W1S	0h	Reserved
19	EXTSYNCOUT	R-0/W1S	0h	Writing 1 to this bit clears the EXTSYNCOUT bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
18	ESMGENEVT	R-0/W1S	0h	Writing 1 to this bit clears the ESMGENEVT bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
17	ADCSOCB	R-0/W1S	0h	Writing 1 to this bit clears the ADCSOCB bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	ADCSOCA	R-0/W1S	0h	Writing 1 to this bit clears the ADCSOCA bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	RESERVED	R-0/W1S	0h	Reserved
14	RESERVED	R-0/W1S	0h	Reserved
13	ECAP6_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the ECAP6_TRIPOUT bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	ECAP5_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the ECAP5_TRIPOUT bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
11	ECAP4_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the ECAP4_TRIPOUT bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	ECAP3_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the ECAP3_TRIPOUT bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	ECAP2_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the ECAP2_TRIPOUT bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	ECAP1_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the ECAP1_TRIPOUT bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	RESERVED	R-0/W1S	0h	Reserved
6	RESERVED	R-0/W1S	0h	Reserved

Table 20-110. XBARCLR7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	ECAP6_OUT	R-0/W1S	0h	Writing 1 to this bit clears the ECAP6_OUT bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	ECAP5_OUT	R-0/W1S	0h	Writing 1 to this bit clears the ECAP5_OUT bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	ECAP4_OUT	R-0/W1S	0h	Writing 1 to this bit clears the ECAP4_OUT bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	ECAP3_OUT	R-0/W1S	0h	Writing 1 to this bit clears the ECAP3_OUT bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	ECAP2_OUT	R-0/W1S	0h	Writing 1 to this bit clears the ECAP2_OUT bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	ECAP1_OUT	R-0/W1S	0h	Writing 1 to this bit clears the ECAP1_OUT bit in the XBARFLG7 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.26 XBARCLR8 Register (Offset = 11Ch) [Reset = 00000000h]

XBARCLR8 is shown in Figure 20-88 and described in Table 20-111.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-88. XBARCLR8 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	EPWM18_TRIPOUT	EPWM17_TRIPOUT
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
EPWM16_TRIPOUT	EPWM15_TRIPOUT	EPWM14_TRIPOUT	EPWM13_TRIPOUT	EPWM12_TRIPOUT	EPWM11_TRIPOUT	EPWM10_TRIPOUT	EPWM9_TRIPOUT
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
EPWM8_TRIPOUT	EPWM7_TRIPOUT	EPWM6_TRIPOUT	EPWM5_TRIPOUT	EPWM4_TRIPOUT	EPWM3_TRIPOUT	EPWM2_TRIPOUT	EPWM1_TRIPOUT
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-111. XBARCLR8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W1S	0h	Reserved
30	RESERVED	R-0/W1S	0h	Reserved
29	RESERVED	R-0/W1S	0h	Reserved
28	RESERVED	R-0/W1S	0h	Reserved
27	RESERVED	R-0/W1S	0h	Reserved
26	RESERVED	R-0/W1S	0h	Reserved
25	RESERVED	R-0/W1S	0h	Reserved
24	RESERVED	R-0/W1S	0h	Reserved
23	RESERVED	R-0/W1S	0h	Reserved
22	RESERVED	R-0/W1S	0h	Reserved
21	RESERVED	R-0/W1S	0h	Reserved
20	RESERVED	R-0/W1S	0h	Reserved
19	RESERVED	R-0/W1S	0h	Reserved
18	RESERVED	R-0/W1S	0h	Reserved
17	EPWM18_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM18_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	EPWM17_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM17_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	EPWM16_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM16_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-111. XBARCLR8 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	EPWM15_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM15_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
13	EPWM14_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM14_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	EPWM13_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM13_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
11	EPWM12_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM12_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	EPWM11_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM11_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	EPWM10_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM10_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	EPWM9_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM9_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	EPWM8_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM8_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
6	EPWM7_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM7_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
5	EPWM6_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM6_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	EPWM5_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM5_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	EPWM4_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM4_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	EPWM3_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM3_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	EPWM2_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM2_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	EPWM1_TRIPOUT	R-0/W1S	0h	Writing 1 to this bit clears the EPWM1_TRIPOUT bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.27 XBARCLR9 Register (Offset = 120h) [Reset = 00000000h]

XBARCLR9 is shown in Figure 20-89 and described in Table 20-112.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-89. XBARCLR9 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	EPWM18_DEL_TRIP	EPWM17_DEL_TRIP
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
EPWM16_DEL_TRIP	EPWM15_DEL_TRIP	EPWM14_DEL_TRIP	EPWM13_DEL_TRIP	EPWM12_DEL_TRIP	EPWM11_DEL_TRIP	EPWM10_DEL_TRIP	EPWM9_DEL_TRIP
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
EPWM8_DEL_TRIP	EPWM7_DEL_TRIP	EPWM6_DEL_TRIP	EPWM5_DEL_TRIP	EPWM4_DEL_TRIP	EPWM3_DEL_TRIP	EPWM2_DEL_TRIP	EPWM1_DEL_TRIP
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-112. XBARCLR9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W1S	0h	Reserved
30	RESERVED	R-0/W1S	0h	Reserved
29	RESERVED	R-0/W1S	0h	Reserved
28	RESERVED	R-0/W1S	0h	Reserved
27	RESERVED	R-0/W1S	0h	Reserved
26	RESERVED	R-0/W1S	0h	Reserved
25	RESERVED	R-0/W1S	0h	Reserved
24	RESERVED	R-0/W1S	0h	Reserved
23	RESERVED	R-0/W1S	0h	Reserved
22	RESERVED	R-0/W1S	0h	Reserved
21	RESERVED	R-0/W1S	0h	Reserved
20	RESERVED	R-0/W1S	0h	Reserved
19	RESERVED	R-0/W1S	0h	Reserved
18	RESERVED	R-0/W1S	0h	Reserved
17	EPWM18_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM18_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	EPWM17_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM17_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-112. XBARCLR9 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	EPWM16_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM16_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
14	EPWM15_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM15_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
13	EPWM14_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM14_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	EPWM13_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM13_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
11	EPWM12_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM12_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	EPWM11_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM11_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	EPWM10_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM10_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	EPWM9_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM9_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	EPWM8_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM8_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
6	EPWM7_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM7_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
5	EPWM6_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM6_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	EPWM5_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM5_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	EPWM4_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM4_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	EPWM3_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM3_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	EPWM2_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM2_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-112. XBARCLR9 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	EPWM1_DEL_TRIP	R-0/W1S	0h	Writing 1 to this bit clears the EPWM1_DEL_TRIP bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.28 XBARCLR10 Register (Offset = 124h) [Reset = 00000000h]

XBARCLR10 is shown in Figure 20-90 and described in Table 20-113.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-90. XBARCLR10 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	EPWM18_DEL_ACTIVE	EPWM17_DEL_ACTIVE
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
EPWM16_DEL_ACTIVE	EPWM15_DEL_ACTIVE	EPWM14_DEL_ACTIVE	EPWM13_DEL_ACTIVE	EPWM12_DEL_ACTIVE	EPWM11_DEL_ACTIVE	EPWM10_DEL_ACTIVE	EPWM9_DEL_ACTIVE
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
EPWM8_DEL_ACTIVE	EPWM7_DEL_ACTIVE	EPWM6_DEL_ACTIVE	EPWM5_DEL_ACTIVE	EPWM4_DEL_ACTIVE	EPWM3_DEL_ACTIVE	EPWM2_DEL_ACTIVE	EPWM1_DEL_ACTIVE
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-113. XBARCLR10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W1S	0h	Reserved
30	RESERVED	R-0/W1S	0h	Reserved
29	RESERVED	R-0/W1S	0h	Reserved
28	RESERVED	R-0/W1S	0h	Reserved
27	RESERVED	R-0/W1S	0h	Reserved
26	RESERVED	R-0/W1S	0h	Reserved
25	RESERVED	R-0/W1S	0h	Reserved
24	RESERVED	R-0/W1S	0h	Reserved
23	RESERVED	R-0/W1S	0h	Reserved
22	RESERVED	R-0/W1S	0h	Reserved
21	RESERVED	R-0/W1S	0h	Reserved
20	RESERVED	R-0/W1S	0h	Reserved
19	RESERVED	R-0/W1S	0h	Reserved
18	RESERVED	R-0/W1S	0h	Reserved
17	EPWM18_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM18_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	EPWM17_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM17_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-113. XBARCLR10 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	EPWM16_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM16_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
14	EPWM15_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM15_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
13	EPWM14_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM14_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	EPWM13_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM13_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
11	EPWM12_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM12_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	EPWM11_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM11_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	EPWM10_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM10_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	EPWM9_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM9_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	EPWM8_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM8_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
6	EPWM7_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM7_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
5	EPWM6_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM6_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	EPWM5_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM5_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	EPWM4_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM4_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	EPWM3_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM3_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-113. XBARCLR10 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	EPWM2_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM2_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	EPWM1_DEL_ACTIVE	R-0/W1S	0h	Writing 1 to this bit clears the EPWM1_DEL_ACTIVE bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.29 XBARCLR11 Register (Offset = 128h) [Reset = 00000000h]

XBARCLR11 is shown in Figure 20-91 and described in Table 20-114.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-91. XBARCLR11 Register

31	30	29	28	27	26	25	24
EPWM16_B0_sclk	EPWM16_A0_sclk	EPWM15_B0_sclk	EPWM15_A0_sclk	EPWM14_B0_sclk	EPWM14_A0_sclk	EPWM13_B0_sclk	EPWM13_A0_sclk
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
EPWM12_B0_sclk	EPWM12_A0_sclk	EPWM11_B0_sclk	EPWM11_A0_sclk	EPWM10_B0_sclk	EPWM10_A0_sclk	EPWM9_B0_sclk	EPWM9_A0_sclk
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
EPWM8_B0_sclk	EPWM8_A0_sclk	EPWM7_B0_sclk	EPWM7_A0_sclk	EPWM6_B0_sclk	EPWM6_A0_sclk	EPWM5_B0_sclk	EPWM5_A0_sclk
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
EPWM4_B0_sclk	EPWM4_A0_sclk	EPWM3_B0_sclk	EPWM3_A0_sclk	EPWM2_B0_sclk	EPWM2_A0_sclk	EPWM1_B0_sclk	EPWM1_A0_sclk
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-114. XBARCLR11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	EPWM16_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM16_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
30	EPWM16_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM16_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
29	EPWM15_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM15_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
28	EPWM15_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM15_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
27	EPWM14_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM14_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
26	EPWM14_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM14_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
25	EPWM13_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM13_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
24	EPWM13_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM13_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-114. XBARCLR11 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	EPWM12_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM12_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
22	EPWM12_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM12_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
21	EPWM11_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM11_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
20	EPWM11_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM11_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
19	EPWM10_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM10_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
18	EPWM10_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM10_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
17	EPWM9_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM9_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	EPWM9_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM9_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	EPWM8_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM8_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
14	EPWM8_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM8_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
13	EPWM7_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM7_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	EPWM7_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM7_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
11	EPWM6_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM6_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	EPWM6_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM6_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	EPWM5_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM5_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	EPWM5_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM5_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	EPWM4_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM4_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-114. XBARCLR11 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	EPWM4_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM4_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
5	EPWM3_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM3_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	EPWM3_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM3_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	EPWM2_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM2_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	EPWM2_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM2_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	EPWM1_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM1_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	EPWM1_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM1_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.30 XBARCLR12 Register (Offset = 12Ch) [Reset = 00000000h]

XBARCLR12 is shown in Figure 20-92 and described in Table 20-115.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-92. XBARCLR12 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	EPWM18_B0_sclk	EPWM18_A0_sclk	EPWM17_B0_sclk	EPWM17_A0_sclk
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-115. XBARCLR12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W1S	0h	Reserved
30	RESERVED	R-0/W1S	0h	Reserved
29	RESERVED	R-0/W1S	0h	Reserved
28	RESERVED	R-0/W1S	0h	Reserved
27	RESERVED	R-0/W1S	0h	Reserved
26	RESERVED	R-0/W1S	0h	Reserved
25	RESERVED	R-0/W1S	0h	Reserved
24	RESERVED	R-0/W1S	0h	Reserved
23	RESERVED	R-0/W1S	0h	Reserved
22	RESERVED	R-0/W1S	0h	Reserved
21	RESERVED	R-0/W1S	0h	Reserved
20	RESERVED	R-0/W1S	0h	Reserved
19	RESERVED	R-0/W1S	0h	Reserved
18	RESERVED	R-0/W1S	0h	Reserved
17	RESERVED	R-0/W1S	0h	Reserved
16	RESERVED	R-0/W1S	0h	Reserved
15	RESERVED	R-0/W1S	0h	Reserved
14	RESERVED	R-0/W1S	0h	Reserved
13	RESERVED	R-0/W1S	0h	Reserved
12	RESERVED	R-0/W1S	0h	Reserved
11	RESERVED	R-0/W1S	0h	Reserved
10	RESERVED	R-0/W1S	0h	Reserved
9	RESERVED	R-0/W1S	0h	Reserved

Table 20-115. XBARCLR12 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R-0/W1S	0h	Reserved
7	RESERVED	R-0/W1S	0h	Reserved
6	RESERVED	R-0/W1S	0h	Reserved
5	RESERVED	R-0/W1S	0h	Reserved
4	RESERVED	R-0/W1S	0h	Reserved
3	EPWM18_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM18_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	EPWM18_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM18_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	EPWM17_B0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM17_B0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	EPWM17_A0_sclk	R-0/W1S	0h	Writing 1 to this bit clears the EPWM17_A0_sclk bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.31 XBARCLR13 Register (Offset = 130h) [Reset = 00000000h]

XBARCLR13 is shown in Figure 20-93 and described in Table 20-116.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-93. XBARCLR13 Register

31	30	29	28	27	26	25	24
MDL16_OUTB	MDL16_OUTA	MDL15_OUTB	MDL15_OUTA	MDL14_OUTB	MDL14_OUTA	MDL13_OUTB	MDL13_OUTA
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
MDL12_OUTB	MDL12_OUTA	MDL11_OUTB	MDL11_OUTA	MDL10_OUTB	MDL10_OUTA	MDL9_OUTB	MDL9_OUTA
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
MDL8_OUTB	MDL8_OUTA	MDL7_OUTB	MDL7_OUTA	MDL6_OUTB	MDL6_OUTA	MDL5_OUTB	MDL5_OUTA
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
MDL4_OUTB	MDL4_OUTA	MDL3_OUTB	MDL3_OUTA	MDL2_OUTB	MDL2_OUTA	MDL1_OUTB	MDL1_OUTA
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-116. XBARCLR13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MDL16_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL16_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
30	MDL16_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL16_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
29	MDL15_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL15_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
28	MDL15_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL15_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
27	MDL14_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL14_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
26	MDL14_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL14_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
25	MDL13_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL13_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
24	MDL13_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL13_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
23	MDL12_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL12_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-116. XBARCLR13 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	MDL12_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL12_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
21	MDL11_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL11_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
20	MDL11_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL11_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
19	MDL10_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL10_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
18	MDL10_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL10_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
17	MDL9_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL9_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	MDL9_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL9_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	MDL8_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL8_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
14	MDL8_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL8_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
13	MDL7_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL7_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	MDL7_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL7_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
11	MDL6_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL6_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	MDL6_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL6_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	MDL5_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL5_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	MDL5_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL5_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	MDL4_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL4_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
6	MDL4_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL4_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-116. XBARCLR13 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	MDL3_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL3_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	MDL3_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL3_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	MDL2_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL2_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	MDL2_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL2_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	MDL1_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL1_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	MDL1_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL1_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.32 XBARCLR14 Register (Offset = 134h) [Reset = 00000000h]

XBARCLR14 is shown in Figure 20-94 and described in Table 20-117.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-94. XBARCLR14 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	MDL18_OUTB	MDL18_OUTA	MDL17_OUTB	MDL17_OUTA
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-117. XBARCLR14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W1S	0h	Reserved
30	RESERVED	R-0/W1S	0h	Reserved
29	RESERVED	R-0/W1S	0h	Reserved
28	RESERVED	R-0/W1S	0h	Reserved
27	RESERVED	R-0/W1S	0h	Reserved
26	RESERVED	R-0/W1S	0h	Reserved
25	RESERVED	R-0/W1S	0h	Reserved
24	RESERVED	R-0/W1S	0h	Reserved
23	RESERVED	R-0/W1S	0h	Reserved
22	RESERVED	R-0/W1S	0h	Reserved
21	RESERVED	R-0/W1S	0h	Reserved
20	RESERVED	R-0/W1S	0h	Reserved
19	RESERVED	R-0/W1S	0h	Reserved
18	RESERVED	R-0/W1S	0h	Reserved
17	RESERVED	R-0/W1S	0h	Reserved
16	RESERVED	R-0/W1S	0h	Reserved
15	RESERVED	R-0/W1S	0h	Reserved
14	RESERVED	R-0/W1S	0h	Reserved
13	RESERVED	R-0/W1S	0h	Reserved
12	RESERVED	R-0/W1S	0h	Reserved
11	RESERVED	R-0/W1S	0h	Reserved
10	RESERVED	R-0/W1S	0h	Reserved
9	RESERVED	R-0/W1S	0h	Reserved

Table 20-117. XBARCLR14 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R-0/W1S	0h	Reserved
7	RESERVED	R-0/W1S	0h	Reserved
6	RESERVED	R-0/W1S	0h	Reserved
5	RESERVED	R-0/W1S	0h	Reserved
4	RESERVED	R-0/W1S	0h	Reserved
3	MDL18_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL18_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	MDL18_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL18_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	MDL17_OUTB	R-0/W1S	0h	Writing 1 to this bit clears the MDL17_OUTB bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	MDL17_OUTA	R-0/W1S	0h	Writing 1 to this bit clears the MDL17_OUTA bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.33 XBARCLR15 Register (Offset = 138h) [Reset = 00000000h]

XBARCLR15 is shown in Figure 20-95 and described in Table 20-118.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-95. XBARCLR15 Register

31	30	29	28	27	26	25	24
CLB6_OUT1	CLB6_OUT0	CLB5_OUT7	CLB5_OUT6	CLB5_OUT3	CLB5_OUT2	CLB5_OUT1	CLB5_OUT0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
CLB4_OUT7	CLB4_OUT6	CLB4_OUT3	CLB4_OUT2	CLB4_OUT1	CLB4_OUT0	CLB3_OUT7	CLB3_OUT6
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
CLB3_OUT3	CLB3_OUT2	CLB3_OUT1	CLB3_OUT0	CLB2_OUT7	CLB2_OUT6	CLB2_OUT3	CLB2_OUT2
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
CLB2_OUT1	CLB2_OUT0	CLB1_OUT7	CLB1_OUT6	CLB1_OUT3	CLB1_OUT2	CLB1_OUT1	CLB1_OUT0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-118. XBARCLR15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CLB6_OUT1	R-0/W1S	0h	Writing 1 to this bit clears the CLB6_OUT1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
30	CLB6_OUT0	R-0/W1S	0h	Writing 1 to this bit clears the CLB6_OUT0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
29	CLB5_OUT7	R-0/W1S	0h	Writing 1 to this bit clears the CLB5_OUT7 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
28	CLB5_OUT6	R-0/W1S	0h	Writing 1 to this bit clears the CLB5_OUT6 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
27	CLB5_OUT3	R-0/W1S	0h	Writing 1 to this bit clears the CLB5_OUT3 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
26	CLB5_OUT2	R-0/W1S	0h	Writing 1 to this bit clears the CLB5_OUT2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
25	CLB5_OUT1	R-0/W1S	0h	Writing 1 to this bit clears the CLB5_OUT1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
24	CLB5_OUT0	R-0/W1S	0h	Writing 1 to this bit clears the CLB5_OUT0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
23	CLB4_OUT7	R-0/W1S	0h	Writing 1 to this bit clears the CLB4_OUT7 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-118. XBARCLR15 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	CLB4_OUT6	R-0/W1S	0h	Writing 1 to this bit clears the CLB4_OUT6 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
21	CLB4_OUT3	R-0/W1S	0h	Writing 1 to this bit clears the CLB4_OUT3 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
20	CLB4_OUT2	R-0/W1S	0h	Writing 1 to this bit clears the CLB4_OUT2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
19	CLB4_OUT1	R-0/W1S	0h	Writing 1 to this bit clears the CLB4_OUT1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
18	CLB4_OUT0	R-0/W1S	0h	Writing 1 to this bit clears the CLB4_OUT0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
17	CLB3_OUT7	R-0/W1S	0h	Writing 1 to this bit clears the CLB3_OUT7 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	CLB3_OUT6	R-0/W1S	0h	Writing 1 to this bit clears the CLB3_OUT6 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	CLB3_OUT3	R-0/W1S	0h	Writing 1 to this bit clears the CLB3_OUT3 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
14	CLB3_OUT2	R-0/W1S	0h	Writing 1 to this bit clears the CLB3_OUT2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
13	CLB3_OUT1	R-0/W1S	0h	Writing 1 to this bit clears the CLB3_OUT1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	CLB3_OUT0	R-0/W1S	0h	Writing 1 to this bit clears the CLB3_OUT0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
11	CLB2_OUT7	R-0/W1S	0h	Writing 1 to this bit clears the CLB2_OUT7 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	CLB2_OUT6	R-0/W1S	0h	Writing 1 to this bit clears the CLB2_OUT6 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	CLB2_OUT3	R-0/W1S	0h	Writing 1 to this bit clears the CLB2_OUT3 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	CLB2_OUT2	R-0/W1S	0h	Writing 1 to this bit clears the CLB2_OUT2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	CLB2_OUT1	R-0/W1S	0h	Writing 1 to this bit clears the CLB2_OUT1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
6	CLB2_OUT0	R-0/W1S	0h	Writing 1 to this bit clears the CLB2_OUT0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-118. XBARCLR15 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	CLB1_OUT7	R-0/W1S	0h	Writing 1 to this bit clears the CLB1_OUT7 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	CLB1_OUT6	R-0/W1S	0h	Writing 1 to this bit clears the CLB1_OUT6 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	CLB1_OUT3	R-0/W1S	0h	Writing 1 to this bit clears the CLB1_OUT3 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	CLB1_OUT2	R-0/W1S	0h	Writing 1 to this bit clears the CLB1_OUT2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	CLB1_OUT1	R-0/W1S	0h	Writing 1 to this bit clears the CLB1_OUT1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	CLB1_OUT0	R-0/W1S	0h	Writing 1 to this bit clears the CLB1_OUT0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.34 XBARCLR16 Register (Offset = 13Ch) [Reset = 00000000h]

XBARCLR16 is shown in Figure 20-96 and described in Table 20-119.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-96. XBARCLR16 Register

31	30	29	28	27	26	25	24
EPG1_EPGOUT3	EPG1_EPGOUT2	EPG1_EPGOUT1	EPG1_EPGOUT0	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
ADCE_EXTMUXSEL3	ADCE_EXTMUXSEL2	ADCE_EXTMUXSEL1	ADCE_EXTMUXSEL0	ADCD_EXTMUXSEL3	ADCD_EXTMUXSEL2	ADCD_EXTMUXSEL1	ADCD_EXTMUXSEL0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
ADCC_EXTMUXSEL3	ADCC_EXTMUXSEL2	ADCC_EXTMUXSEL1	ADCC_EXTMUXSEL0	ADCB_EXTMUXSEL3	ADCB_EXTMUXSEL2	ADCB_EXTMUXSEL1	ADCB_EXTMUXSEL0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
ADCA_EXTMUXSEL3	ADCA_EXTMUXSEL2	ADCA_EXTMUXSEL1	ADCA_EXTMUXSEL0	CLB6_OUT7	CLB6_OUT6	CLB6_OUT3	CLB6_OUT2
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-119. XBARCLR16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	EPG1_EPGOUT3	R-0/W1S	0h	Writing 1 to this bit clears the EPG1_EPGOUT3 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
30	EPG1_EPGOUT2	R-0/W1S	0h	Writing 1 to this bit clears the EPG1_EPGOUT2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
29	EPG1_EPGOUT1	R-0/W1S	0h	Writing 1 to this bit clears the EPG1_EPGOUT1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
28	EPG1_EPGOUT0	R-0/W1S	0h	Writing 1 to this bit clears the EPG1_EPGOUT0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
27	RESERVED	R-0/W1S	0h	Reserved
26	RESERVED	R-0/W1S	0h	Reserved
25	RESERVED	R-0/W1S	0h	Reserved
24	RESERVED	R-0/W1S	0h	Reserved
23	ADCE_EXTMUXSEL3	R-0/W1S	0h	Writing 1 to this bit clears the ADCE_EXTMUXSEL3 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
22	ADCE_EXTMUXSEL2	R-0/W1S	0h	Writing 1 to this bit clears the ADCE_EXTMUXSEL2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-119. XBARCLR16 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	ADCE_EXTMUXSEL1	R-0/W1S	0h	Writing 1 to this bit clears the ADCE_EXTMUXSEL1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
20	ADCE_EXTMUXSEL0	R-0/W1S	0h	Writing 1 to this bit clears the ADCE_EXTMUXSEL0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
19	ADCD_EXTMUXSEL3	R-0/W1S	0h	Writing 1 to this bit clears the ADCD_EXTMUXSEL3 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
18	ADCD_EXTMUXSEL2	R-0/W1S	0h	Writing 1 to this bit clears the ADCD_EXTMUXSEL2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
17	ADCD_EXTMUXSEL1	R-0/W1S	0h	Writing 1 to this bit clears the ADCD_EXTMUXSEL1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	ADCD_EXTMUXSEL0	R-0/W1S	0h	Writing 1 to this bit clears the ADCD_EXTMUXSEL0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	ADCC_EXTMUXSEL3	R-0/W1S	0h	Writing 1 to this bit clears the ADCC_EXTMUXSEL3 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
14	ADCC_EXTMUXSEL2	R-0/W1S	0h	Writing 1 to this bit clears the ADCC_EXTMUXSEL2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
13	ADCC_EXTMUXSEL1	R-0/W1S	0h	Writing 1 to this bit clears the ADCC_EXTMUXSEL1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	ADCC_EXTMUXSEL0	R-0/W1S	0h	Writing 1 to this bit clears the ADCC_EXTMUXSEL0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
11	ADCB_EXTMUXSEL3	R-0/W1S	0h	Writing 1 to this bit clears the ADCB_EXTMUXSEL3 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	ADCB_EXTMUXSEL2	R-0/W1S	0h	Writing 1 to this bit clears the ADCB_EXTMUXSEL2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	ADCB_EXTMUXSEL1	R-0/W1S	0h	Writing 1 to this bit clears the ADCB_EXTMUXSEL1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	ADCB_EXTMUXSEL0	R-0/W1S	0h	Writing 1 to this bit clears the ADCB_EXTMUXSEL0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-119. XBARCLR16 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	ADCA_EXTMUXSEL3	R-0/W1S	0h	Writing 1 to this bit clears the ADCA_EXTMUXSEL3 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
6	ADCA_EXTMUXSEL2	R-0/W1S	0h	Writing 1 to this bit clears the ADCA_EXTMUXSEL2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
5	ADCA_EXTMUXSEL1	R-0/W1S	0h	Writing 1 to this bit clears the ADCA_EXTMUXSEL1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	ADCA_EXTMUXSEL0	R-0/W1S	0h	Writing 1 to this bit clears the ADCA_EXTMUXSEL0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	CLB6_OUT7	R-0/W1S	0h	Writing 1 to this bit clears the CLB6_OUT7 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	CLB6_OUT6	R-0/W1S	0h	Writing 1 to this bit clears the CLB6_OUT6 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	CLB6_OUT3	R-0/W1S	0h	Writing 1 to this bit clears the CLB6_OUT3 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	CLB6_OUT2	R-0/W1S	0h	Writing 1 to this bit clears the CLB6_OUT2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.35 XBARCLR17 Register (Offset = 140h) [Reset = 00000000h]

XBARCLR17 is shown in Figure 20-97 and described in Table 20-120.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-97. XBARCLR17 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						MCANF_FEVT2	MCANF_FEVT1
R-0h						R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
MCANF_FEVT0	MCANE_FEVT2	MCANE_FEVT1	MCANE_FEVT0	MCAND_FEVT2	MCAND_FEVT1	MCAND_FEVT0	MCANC_FEVT2
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
MCANC_FEVT1	MCANC_FEVT0	MCANB_FEVT2	MCANB_FEVT1	MCANB_FEVT0	MCANA_FEVT2	MCANA_FEVT1	MCANA_FEVT0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-120. XBARCLR17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17	MCANF_FEVT2	R-0/W1S	0h	Writing 1 to this bit clears the MCANF_FEVT2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	MCANF_FEVT1	R-0/W1S	0h	Writing 1 to this bit clears the MCANF_FEVT1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	MCANF_FEVT0	R-0/W1S	0h	Writing 1 to this bit clears the MCANF_FEVT0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
14	MCANE_FEVT2	R-0/W1S	0h	Writing 1 to this bit clears the MCANE_FEVT2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
13	MCANE_FEVT1	R-0/W1S	0h	Writing 1 to this bit clears the MCANE_FEVT1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	MCANE_FEVT0	R-0/W1S	0h	Writing 1 to this bit clears the MCANE_FEVT0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
11	MCAND_FEVT2	R-0/W1S	0h	Writing 1 to this bit clears the MCAND_FEVT2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	MCAND_FEVT1	R-0/W1S	0h	Writing 1 to this bit clears the MCAND_FEVT1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 20-120. XBARCLR17 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	MCAND_FEVT0	R-0/W1S	0h	Writing 1 to this bit clears the MCAND_FEVT0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	MCANC_FEVT2	R-0/W1S	0h	Writing 1 to this bit clears the MCANC_FEVT2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	MCANC_FEVT1	R-0/W1S	0h	Writing 1 to this bit clears the MCANC_FEVT1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
6	MCANC_FEVT0	R-0/W1S	0h	Writing 1 to this bit clears the MCANC_FEVT0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
5	MCANB_FEVT2	R-0/W1S	0h	Writing 1 to this bit clears the MCANB_FEVT2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	MCANB_FEVT1	R-0/W1S	0h	Writing 1 to this bit clears the MCANB_FEVT1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	MCANB_FEVT0	R-0/W1S	0h	Writing 1 to this bit clears the MCANB_FEVT0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	MCANA_FEVT2	R-0/W1S	0h	Writing 1 to this bit clears the MCANA_FEVT2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	MCANA_FEVT1	R-0/W1S	0h	Writing 1 to this bit clears the MCANA_FEVT1 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	MCANA_FEVT0	R-0/W1S	0h	Writing 1 to this bit clears the MCANA_FEVT0 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

20.5.9.36 XBARCLR18 Register (Offset = 144h) [Reset = 00000000h]

XBARCLR18 is shown in Figure 20-98 and described in Table 20-121.

Return to the Summary Table.

This register is used to clear the flag(s) in the XBARFLG register.
1: Clears the corresponding bit in the XBARFLG register.
0: Writing 0 has no effect

Figure 20-98. XBARCLR18 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				CPU3ERADEVT11	CPU3ERADEVT10	CPU3ERADEVT9	CPU3ERADEVT8
R-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
CPU2ERADEVT11	CPU2ERADEVT10	CPU2ERADEVT9	CPU2ERADEVT8	CPU1ERADEVT11	CPU1ERADEVT10	CPU1ERADEVT9	CPU1ERADEVT8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
FSIRXD_TRIG_3	FSIRXD_TRIG_2	FSIRXC_TRIG_3	FSIRXC_TRIG_2	FSIRXB_TRIG_3	FSIRXB_TRIG_2	FSIRXA_TRIG_3	FSIRXA_TRIG_2
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 20-121. XBARCLR18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	CPU3ERADEVT11	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU3ERADEVT11 input was triggered 0: CPU3ERADEVT11 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
18	CPU3ERADEVT10	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU3ERADEVT10 input was triggered 0: CPU3ERADEVT10 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
17	CPU3ERADEVT9	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU3ERADEVT9 input was triggered 0: CPU3ERADEVT9 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-121. XBARCLR18 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CPU3ERADEVT8	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU3ERADEVT8 input was triggered 0: CPU3ERADEVT8 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	CPU2ERADEVT11	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU2ERADEVT11 input was triggered 0: CPU2ERADEVT11 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	CPU2ERADEVT10	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU2ERADEVT10 input was triggered 0: CPU2ERADEVT10 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
13	CPU2ERADEVT9	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU2ERADEVT9 input was triggered 0: CPU2ERADEVT9 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
12	CPU2ERADEVT8	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU2ERADEVT8 input was triggered 0: CPU2ERADEVT8 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
11	CPU1ERADEVT11	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU1ERADEVT11 input was triggered 0: CPU1ERADEVT11 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	CPU1ERADEVT10	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU1ERADEVT10 input was triggered 0: CPU1ERADEVT10 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	CPU1ERADEVT9	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU1ERADEVT9 input was triggered 0: CPU1ERADEVT9 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-121. XBARCLR18 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	CPU1ERADEVT8	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: CPU1ERADEVT8 input was triggered 0: CPU1ERADEVT8 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	FSIRXD_TRIG_3	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIRXD_TRIG_3 input was triggered 0: FSIRXD_TRIG_3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	FSIRXD_TRIG_2	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIRXD_TRIG_2 input was triggered 0: FSIRXD_TRIG_2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	FSIRXC_TRIG_3	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIRXC_TRIG_3 input was triggered 0: FSIRXC_TRIG_3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	FSIRXC_TRIG_2	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIRXC_TRIG_2 input was triggered 0: FSIRXC_TRIG_2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
3	FSIRXB_TRIG_3	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIRXB_TRIG_3 input was triggered 0: FSIRXB_TRIG_3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	FSIRXB_TRIG_2	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIRXB_TRIG_2 input was triggered 0: FSIRXB_TRIG_2 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	FSIRXA_TRIG_3	R-0/W1S	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: FSIRXA_TRIG_3 input was triggered 0: FSIRXA_TRIG_3 Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 20-121. XBARCLR18 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	FSIRXA_TRIG_2	R-0/W1S	0h	Writing 1 to this bit clears the FSIRXA_TRIG_2 bit in this register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Chapter 21

Embedded Pattern Generator (EPG)

This chapter describes the Embedded Pattern Generator (EPG) module.

21.1 Introduction

The Embedded Pattern Generator (EPG) module is a customizable pattern and clock generator that can serve many test and application scenarios that require a simple pattern generator or a periodic clock generator. The EPG module can also be used to capture an incoming serial stream of data.

21.1.1 Features

Features of the EPG module are:

- Clock generation:
  - Independent clock generation and clock division
  - Synchronous clock generation with programmable offsets
  - Can generate clocks to act as source for external modulator and internal demodulator for the SDFM
- Pattern generation:
  - Independent serial data stream generation
  - Serial data stream and the associated clock generation
  - Ability to skew clock with respect to serial data
  - Synchronous data stream with programmable offset with respect to one another
  - Can generate waveforms for loopback test of communication peripherals
  - Useful as diagnostics test if not already built-in

The EPG output signals are connected to GPIOs or other peripherals inside the device. The EPG inputs act as shift register inputs to capture incoming serial data streams. This allows the EPG to be configured as a custom serial module. The EPG is also capable of generating interrupts that are used to supply the pattern generators with new data or signal the completion of a generated pattern.

21.1.2 EPG Block Diagram

Figure 21-1 shows the EPG overview block diagram.

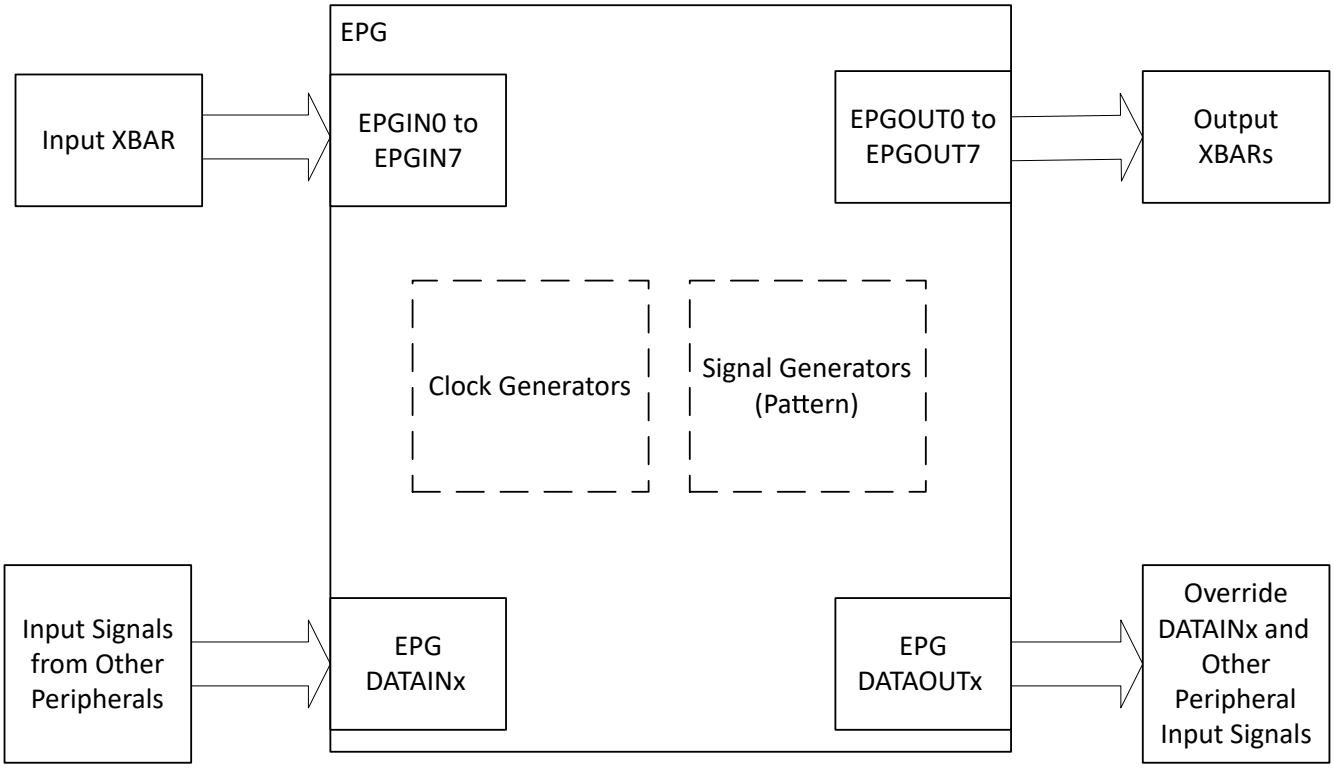


Figure 21-1. EPG Overview Block Diagram

Figure 21-2 shows the EPG block diagram.

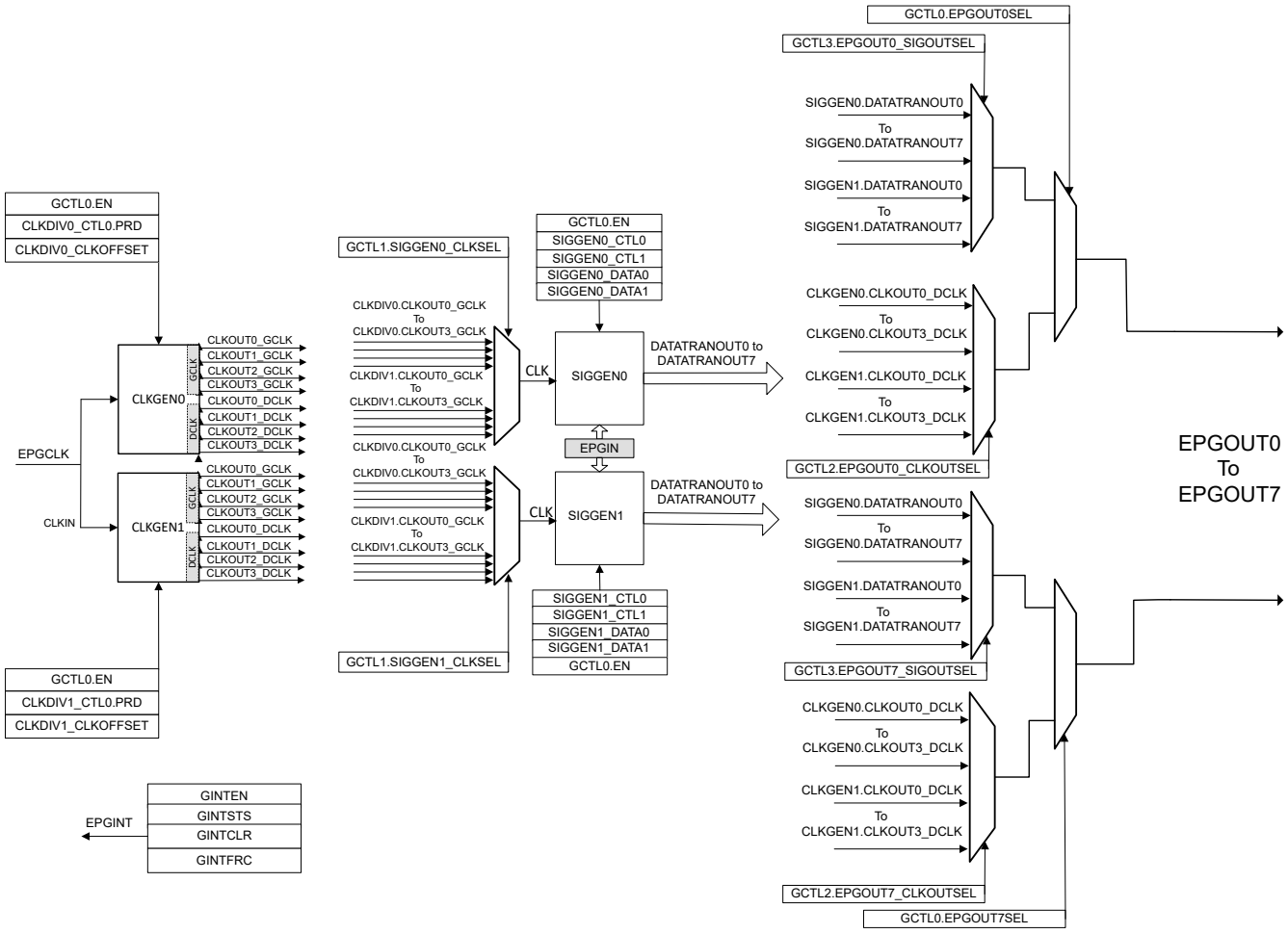


Figure 21-2. EPG Detailed Block Diagram

Note

The number of available EPGx SIGGEN modules for each device can be different. This diagram shows an EPG with 2 SIGGEN modules (SIGGEN0 and SIGGEN1). Refer to the EPG_REGS register description to see how many SIGGEN modules are available for a specific device.

The EPG input clock (EPGCLK) is sourced from PERx.SYSCLK.

21.1.3 EPG Related Collateral

Foundational Materials

- C2000 Embedded Pattern Generator (Video)

Getting Started Materials

- *Designing With the C2000™ Embedded Pattern Generator (EPG) Application Report*

21.2 Clock Generator Modules

The clock generator modules use the EPG input clock and generate four output clocks, see Figure 21-3. Each of the four output clocks (CLKOUT0 to CLKOUT3) has a DCLK and a GCLK version. The EPG clock division results in a gated, GCLK, version of the EPG input clock which are named CLKOUT0_GCLK to CLKOUT3_GCLK. The other version, DCLK (CLKOUT0_DCLK to CLKOUT3_DCLK), have the same period but with an approximately 50% duty cycle. The clock generation takes place using the divider settings CLKDIVx_CTL0.PRD, and clock offset settings CLKDIVx_CLK.CLKyOFFSET. The RUNCLOCK signal generated by the clock stop logic determines when the clock generation is started and stopped.

The clock divider counter is a simple up counter, which has a period determined by CLKDIVx_CTL0.PRD. This divider counter begins counting when RUNCLOCK is set. The CLKOUT0 to CLKOUT4 GCLKs are gated versions of the CLKIN (EPG input clock). The clock gates are enabled when the counter value matches the corresponding clock offsets. In the case where RUNCLOCK is cleared, the clock gates are disabled and the counter is set to zero.

Note

The CLKDIVx_CTL0.PRD register can only be written when GCTL0.EN is 0.

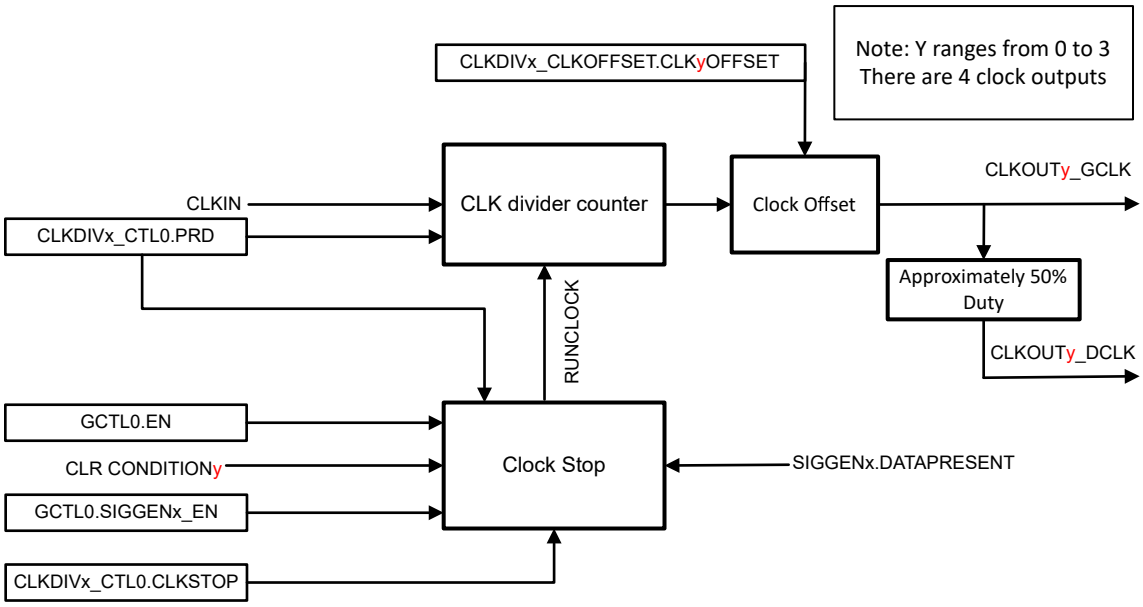


Figure 21-3. EPG Clock Generator

21.2.1 DCLK (50% duty cycle clock)

The CLKOUTy_DCLK is generated by setting the clock output for half the clock divider period. When CLKDIVx_CTL0.PRD is set to zero, CLKOUTy_DCLK is the same as the input clock.

21.2.2 Clock Stop

Figure 21-4 shows how the Clock Stop module sets and clears the RUNCLOCK signal.

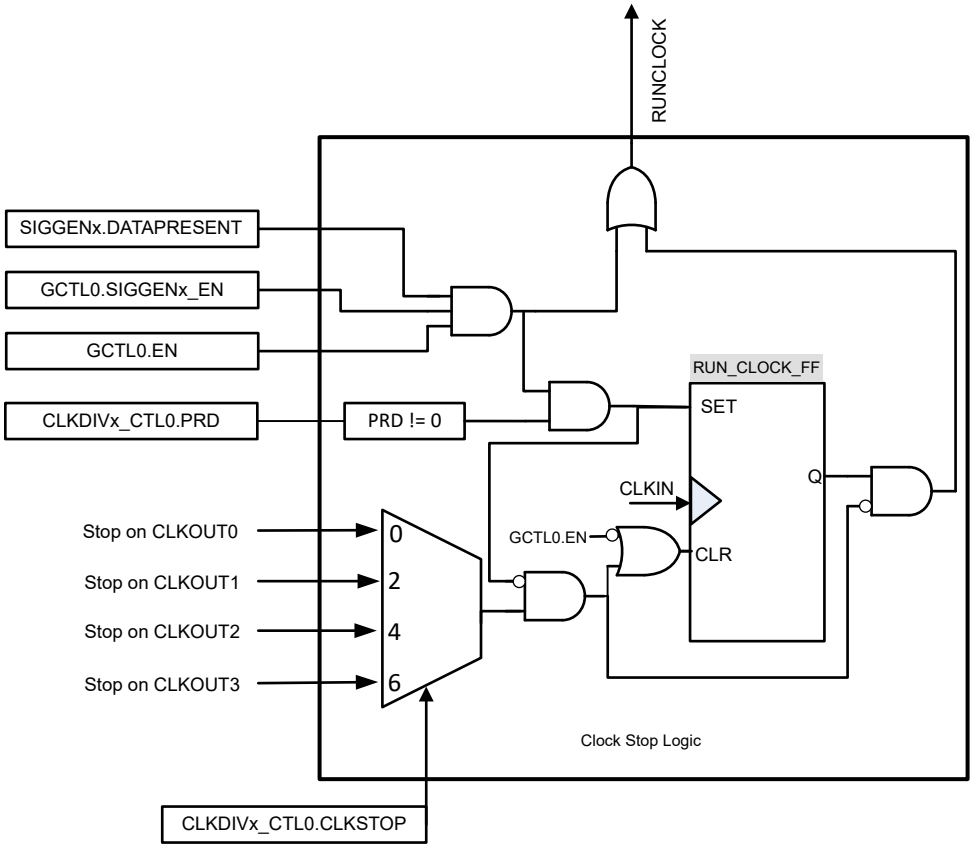


Figure 21-4. EPG Clock Stop

In signal generator modes, GCTL0.SIGGENx_EN bit is cleared when BIT_LENGTH shifts (or rotates) is completed. This makes sure that data is not shifted out. In addition, clock generation (generation of CLKOUT0 to CLKOUT3) also is stopped to make sure that sampling of input data does not continue after BIT_LENGTH shifts when GCTL0.SIGGENx_EN is cleared to 0.

The RUNCLOCK signal has to be high for the clock generation circuitry to be active. When GCTL0.SIGGENx_EN is cleared, the clock generation can be selected to stop on the falling edge of CLKOUT0 to CLKOUT3.

The clock stop module operates as follow:

- RUNCLOCK is asserted as soon as both GCTL0.SIGGENx_EN and GCTL0.EN are set.
- RUN_CLOCK_FF is set when both GCTL0.SIGGENx_EN and GCTL0.EN are set and CLKDIVx_CTL0.PRD is not zero.
- When GCTL0.SIGGENx_EN is cleared, RUN_CLOCK_FF is not cleared immediately. This makes sure that RUNCLOCK remains set and the clock generation circuitry remains enabled.
- When SIGGENx.DATA0 and SIGGENx.DATA1 are not filled with data before the next shift sequence (repeat shift modes) and signal generator is waiting for data to be written, clocks are stopped.
- The RUN_CLOCK_FF can be cleared on the falling edge of CLKOUT0 to CLKOUT3 based on the configuration of CLKSTOP field.
- Clearing GCTL0.EN clears RUN_CLOCK_FF unconditionally.

21.3 Signal Generator Module

The signal generator module is the main component of the EPG and generates the data stream that follows custom patterns. Figure 21-5 shows the main components. This module has 8 output ports DATATRANOUT0 to DATATRANOUT7. The two registers SIGGENx_DATA1 and SIGGENx_DATA0 constitute a 64-bit bus named DATA[63:0]. DATATRANIN[63:0], which is used for all the data transform operations, can be DATA[63:0] or bit reversed DATA (when SIGGENx_CTL0.BRIN bit is set). The DATATRANOUT0 to DATATRANOUT7 are connected to DATATRANIN0 to DATATRANIN7 when the signal generator is not in BIT_BANG mode. In BIT_BANG mode, DATATRANOUT0 to DATATRANOUT7 are connected to DATATRANIN0, DATATRANIN8, and DATATRANIN16 to DATATRANIN56.

In addition to generating data outputs, one of the 8 EPGIN inputs can act as data input to SIGGENx_DATAy registers. In Figure 21-5, the EPGIN_MUX block illustrates the mechanism used to capture the input data stream. This enables one to capture a data input stream using EPG module.

Data transformation is done on the DATATRANIN bus, and is determined by the configured mode (SIGGENx_CTL0.MODE), provided GCTL0.EN and GCTL0.SIGGENx_EN are both set. If either of these enable bits are 0, then the data output of the transform block is the same as the input. The transformed output is bit reversed when SIGGENx_CTL0.BROUT bit is set. Conditions under which the DATA active register (SIGGENx_DATA0_ACTIVE and SIGGENx_DATA1_ACTIVE) are:

- Loaded from SIGGENx_DATA1 and SIGGENx_DATA0
- Directly updated on a memory mapped write to SIGGENx_DATA1 and SIGGENx_DATA0
- Holds the current data

Table 21-1. SIGGENx Active Register Loading

Condition	SIGGENx_DATA1_ACTIVE DATA ACTIVE Register [63:32]	SIGGENx_DATA0_ACTIVE DATA ACTIVE Register [31:0]
Memory mapped write to SIGGENx_DATA0 register and GCTL0.SIGGENx_EN is 0	No updates	Updated with the value written to SIGGENx_DATA0 register
Memory mapped write to SIGGENx_DATA1 register and GCTL0.SIGGENx_EN is 0.	Updated with the value written to SIGGENx_DATA1 register	No updates
“BITLENGTH” number of shifts are done, and Mode is SHIFT_RIGHT_REPEAT or SHIFT_LEFT_REPEAT, and BITLENGTH <= 32, and Either SIGGENx_DATA0 or SIGGENx_DATA1 has been updated	Copy SIGGENx_DATA1 register content	Copy SIGGENx_DATA0 register content
“BITLENGTH” number of shifts are done, and Mode is SHIFT_RIGHT_REPEAT or SHIFT_LEFT_REPEAT, and BITLENGTH >= 32, and Both SIGGENx_DATA0 and SIGGENx_DATA1 have been updated	Copy SIGGENx_DATA1 register contents	Copy SIGGENx_DATA0 register contents
“BITLENGTH” number of shifts are done, and Mode is SHIFT_RIGHT_REPEAT or SHIFT_LEFT_REPEAT, and BITLENGTH <= 32, and Neither SIGGENx_DATA0, nor SIGGENx_DATA1 has been updated	Hold the current value, no shifts	Hold the current value, no shifts
“BITLENGTH” number of shifts are done, and Mode is SHIFT_RIGHT_REPEAT or SHIFT_LEFT_REPEAT, and BITLENGTH >= 32, and Neither SIGGENx_DATA0, nor SIGGENx_DATA1 has not been updated	Hold the current value, no shifts	Hold the current value, no shifts
All other conditions	Updates based on current mode of operation	Updates based on current mode of operation

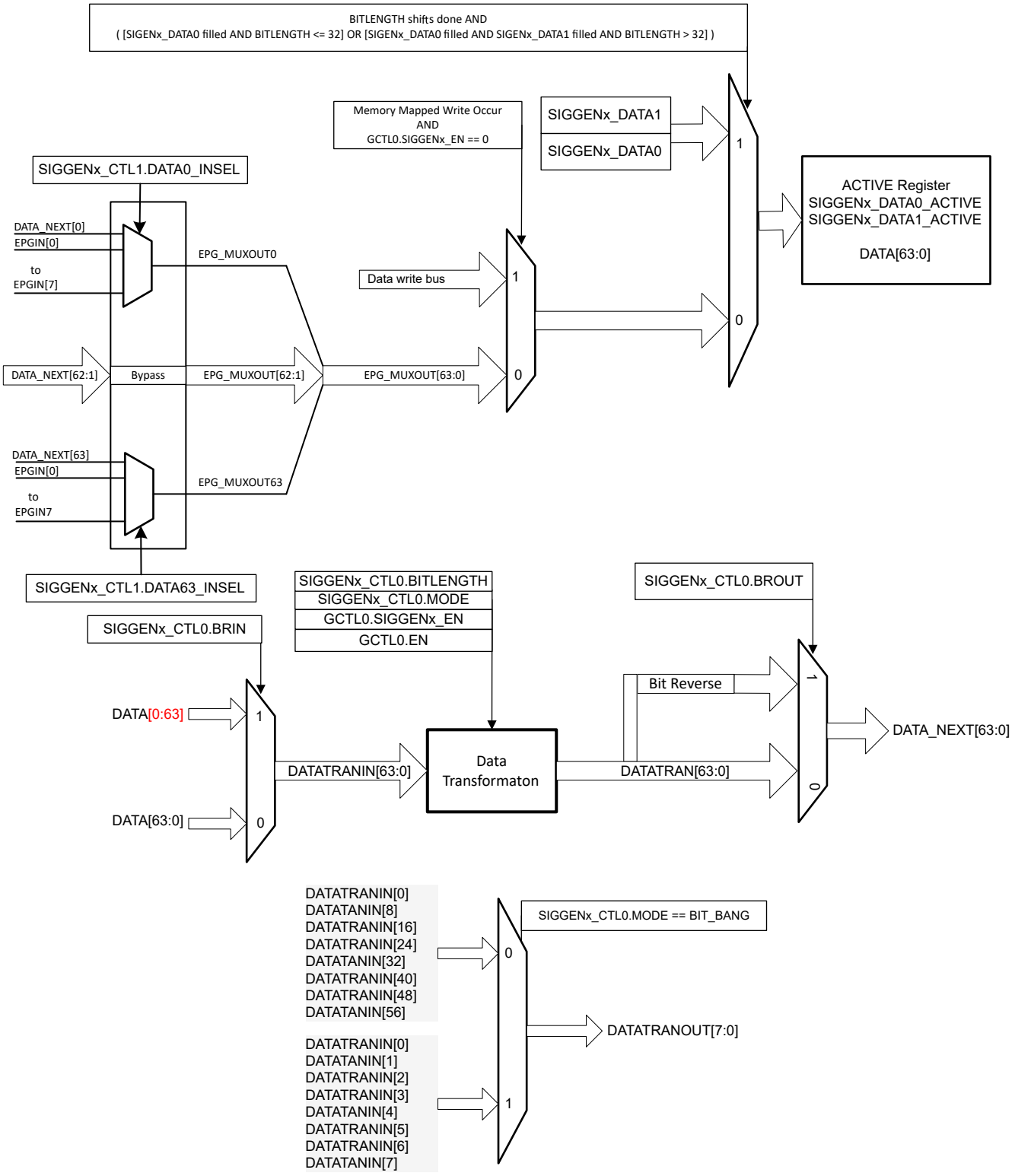


Figure 21-5. EPG Signal Generator Detailed Overview

Following are the possible data transformations:

Bit-bang mode: In this mode, DATATRAN[63:0] is the same as DATATRANIN[63:0].

Shift right once mode: In this mode, DATATRAN[63:0] = {0,DATATRANIN[63:1]}. After SIGENx_CTL0.BITLENGTH shifts, SIGENx_CTL0.EN is cleared.

Shift right repeat mode: In this mode, DATATRAN[63:0] = {0,DATATRANIN[63:1]}. After SIGENx_CTL0.BITLENGTH shifts, load the data as per Table 21-1.

Rotate right once mode: In this mode, DATATRAN[63:0] = {DATATRANIN[0],DATATRANIN[63:1]}. After SIGENx_CTL0.BITLENGTH shifts, active register is loaded from the {DATA1,DATA0} register, and SIGENx_CTL0.EN is cleared.

Rotate right repeat: This mode is same as rotate right once, except that SIGENx_CTL0.EN is not cleared upon SIGENx_CTL0.BITLENGTH rotates.

Shift left once mode: In this mode, DATATRAN[63:0] = {DATATRANIN[62:1],0}. After SIGENx_CTL0.BITLENGTH shifts, SIGENx_CTL0.EN is cleared.

Shift left repeat mode: In this mode, DATATRAN[63:0] = {DATATRANIN[62:1],0}. After SIGENx_CTL0.BITLENGTH shifts, load the data as per Table 21-1.

Rotate left once mode: In this mode, DATATRAN[63:0] = {DATATRANIN[62:1],DATATRANIN[63]}. After SIGENx_CTL0.BITLENGTH shifts, active register is loaded from the {DATA1,DATA0} register, and SIGENx_CTL0.EN is cleared.

Rotate left repeat: This mode is same as rotate left once, except that SIGENx_CTL0.EN is not cleared upon SIGENx_CTL0.BITLENGTH rotates.

Note

SIGGENx_CTL0.MODE and SIGGENx_CTL.BITWIDTH must only be updated when GCTL0.SIGGENx_EN is 0.

21.4 EPG Peripheral Signal Mux Selection

EPG DATAINx signals are connected to input signal sources from other peripherals. The EPG DATAINx signal sources are listed in Table 21-2.

Table 21-2. EPG Data Input Connections

EPG1 Data Connection (DATAIN/DATAOUT)	Signal Name
0	UARTD_RX
1	UARTE_RX
2	LINA_RX
3	LINB_RX
4	I2CA_SDA
5	I2CA_SCL
6	I2CB_SDA
7	I2CB_SCL
8	FSIRXA_D0
9	FSIRXA_D1
10	FSIRXB_D0
11	FSIRXB_D1
12	FSIRXA_CLK
13	FSIRXB_CLK
14	FSIRXC_CLK
15	FSIRXC_D0
16	FSIRXC_D1
17	SPIA_CLK
18	SPIA_PICO
19	SPIA_POCI
20	SPIA_PTE
21	SPIB_CLK
22	SPIB_PICO
23	SPIB_POCI
24	SPIB_PTE
25	SD1_C1
26	SD1_C2
27	SD1_C3
28	SD1_C4
29	SD2_C1
30	SD2_C2
31	SD2_C3
32	SD2_C4
33	SD3_C1
34	SD3_C2

Table 21-2. EPG Data Input Connections (continued)

EPG1 Data Connection (DATAIN/DATAOUT)	Signal Name
35	SD3_C3
36	SD3_C4
37	SD4_C1
38	SD4_C2
39	SD4_C3
40	SD4_C4
41	EQEP1A
42	EQEP1B
43	EQEP1I
44	EQEP1S
45	EQEP2A
46	EQEP2B
47	EQEP2I
48	EQEP2S
49	ECAP1
50	ECAP2
51	ECAP3
52	ECAP4
53	ECAP5
54	ECAP6
55	MCANA_RX
56	MCANB_RX
57	MCANC_RX
58	MCAND_RX
59	MCANE_RX
60	MCANF_RX
61	UARTA_RX
62	UARTB_RX
63	UARTC_RX

EPGOUT0 to EPGOUT7 can override peripheral input signals connected to DATAINx. For example, the peripheral CAN RX input signal can be overwritten by EPGOUTx. DATAINx can be passed to DATAOUTx untouched, if EPG is not required, or the DATAINx can be replaced with EPGOUTy (y = x%8). The EPGMXSEL0 and EPGMXSEL1 registers must be configured to select the source of the DATAOUTx signals. Also, EPGOUTx signals can be connected to GPIOs through the Output XBARs and CLB Output XBARs, while EPGINx signals can be connected to the Input XBARs.

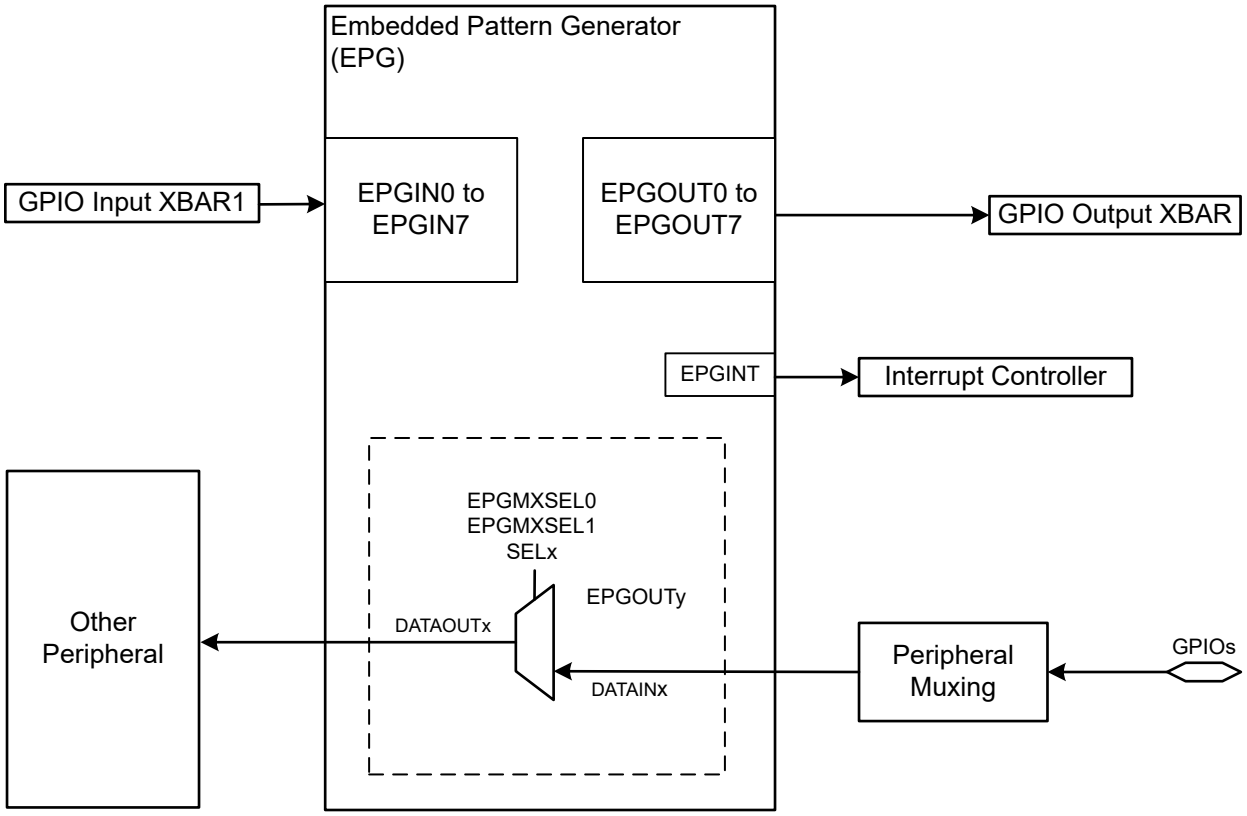


Figure 21-6. EPG Peripheral Signal Muxing

Table 21-3. EPG Input Connections

EPG1 Input Port	Source Signal
0	INPUTXBAR
1	INPUTXBAR
2	INPUTXBAR
3	INPUTXBAR
4-7	Reserved

Table 21-4. EPG Output Connections

EPG1 Output Port	Destination Signal
0	OUTPUTXBAR
1	OUTPUTXBAR
2	OUTPUTXBAR
3	OUTPUTXBAR
4-7	Reserved

21.5 Application Software Notes

The following points are important considerations when utilizing the EPG:

- SIGGENx_CTL0.MODE and SIGGENx_CTL0.BITLENGTH can only be written to when SIGGENx_CTL0.EN is 0
- CLKDIVx_CTL0.PRD register can be written, when GCTL0.EN is 0
- GCTL0.EN can be enabled before GCTL0.SIGGENx_EN

21.6 EPG Example Use Cases

This section provides example register configurations for different EPG use cases.

Note

Some examples can require more than one SIGGEN module. Check the device registers or data sheet for the number of available SIGGEN modules.

21.6.1 EPG Example: Synchronous Clocks with Offset

Example 21-1 register configuration generates 4 clocks, all synchronous to one another with edges offset by 2 clock cycles. In Example 21-1, a clock divide value of 12 is used.

Example 21-1. Synchronous Clocks with Offset Register Configuration

Register	Value	Selected Mode
Epg1MuxRegs		
EPGMXSEL0.SEL0	0x1	Select EPGOUT0 to drive DATAOUT[0]
EPGMXSEL0.SEL1	0x1	Select EPGOUT1 to drive DATAOUT[1]
EPGMXSEL0.SEL2	0x1	Select EPGOUT2 to drive DATAOUT[2]
EPGMXSEL0.SEL3	0x1	Select EPGOUT3 to drive DATAOUT[3]
Epg1Regs		
Global Settings		
GCTL0.EPGOUT0SEL	0x0	Selects signal mux output on EPGOUT0
GCTL0.EPGOUT1SEL	0x0	Selects signal mux output on EPGOUT1
GCTL0.EPGOUT2SEL	0x0	Selects signal mux output on EPGOUT2
GCTL0.EPGOUT3SEL	0x0	Selects signal mux output on EPGOUT3
GCTL3.EPGOUT0_SIGOUTSEL	0x0	Select SIGGEN0.OUT[0] on EPGOUT0
GCTL3.EPGOUT1_SIGOUTSEL	0x1	Select SIGGEN0.OUT[1] on EPGOUT1
GCTL3.EPGOUT2_SIGOUTSEL	0x2	Select SIGGEN0.OUT[2] on EPGOUT2
GCTL3.EPGOUT3_SIGOUTSEL	0x3	Select SIGGEN0.OUT[3] on EPGOUT3
GCTL1.SIGGEN0_CLKSEL	0x0	Select CLKOUT0 of CLKGEN0 as the clock source of SIGGEN0
CLKGEN0 Setting		
CLKDIV0_CTL0.PRD	0x0	Divide by 1
CLKDIV0_CLKOFFSET.CLK0OFFSET	0x0	No offset
SIGGEN0 Mode and Bit Length Configuration		
SIGGEN0_CTL0.BITLENGTH	0xC	Configure bit length to 12 to get a divide-by-12 clock.
SIGGEN0_CTL0.MODE	0x3	Configure the mode to rotate right repeat mode to generate a periodic waveform.
SIGGEN0_DATA0[11:0]	0000 0111 1110	Initialize the 12 bits to 6 ones and 6 zeroes, which makes sure of a 50% duty cycle clock.
SIGGEN0_DATA0[27:16]	0001 1111 1000	Create a 2-cycle offset with respect to first clock.
SIGGEN0_DATA1[11:0]	0111 1110 0000	Create a 4-cycle offset with respect to first clock.
SIGGEN0_DATA1[27:16]	1111 1000 0001	Create a 6-cycle offset with respect to first clock.

21.6.2 EPG Example: Serial Data Bit Stream (LSB first)

Example 21-2 register configuration shifts out a data word, the data rate is set to divide by 8 and the LSB is shifted out first. After 32 shifts are completed, an interrupt is generated for further sequencing.

Example 21-2. Serial Data Bit Stream (LSB first) Register Configuration

Register	Value	Selected Mode
Epg1MuxRegs		
EPGMXSEL0.SEL0	0x1	Select EPGOUT0 to drive DATAOUT[0]
Epg1Regs		
Global Settings		
GCTL1.SIGGEN0_CLKSEL	0x0	Select CLKOUT0 of CLKGEN0 as the clock source of SIGGEN0
CLKGEN0 Setting		
CLKDIV0_CTL0.PRD	0x7	Divide by 8
CLKDIV0_CLKOFFSET.CLK0OFFSET	0x0	No offset
SIGGEN0 Mode and Bit Length Configuration		
SIGGEN0_CTL0.BITLENGTH	0x20	Do 32 shifts.
SIGGEN0_CTL0.MODE	0x1	Configure the mode to shift right once mode. Generates an interrupt after 32 shifts.
SIGGEN0_DATA0[15:0]	0xAA55	Data to be shifted out
SIGGEN0_DATA0[31:16]	0xCCCC	Data to be shifted out
SIGGEN0_DATA1[15:0]	0xAA55	Data to be shifted out
SIGGEN0_DATA1[31:16]	0x55AA	Data to be shifted out

21.6.3 EPG Example: Serial Data Bit Stream (MSB first)

Example 21-3 register configuration shifts out a data word, the data rate is set to divide by 8 and MSB is shifted out first. After 32 shifts are complete, an interrupt is generated for further sequencing.

Example 21-3. Serial Data Bit Stream (MSB first) Register Configuration

Register	Value	Selected Mode
Epg1MuxRegs		
EPGMXSEL0.SEL0	0x1	Select EPGOUT0 to drive DATAOUT[0]
Epg1Regs		
Global Settings		
GCTL0.EPGOUT0SEL	0x0	Selects signal mux output on EPGOUT0
GCTL3.EPGOUT0_SIGOUTSEL	0x4	Select SIGGEN0.OUT[4] on EPGOUT0, on 64-bit reversal, 31 bit appears on 32 bit (hence, configuring to 4).
GCTL1.SIGGEN0_CLKSEL	0x0	Select CLKOUT0 of CLKGEN0 as the clock source of SIGGEN0
CLKGEN0 Setting		
CLKDIV0_CTL0.PRD	0x7	Divide by 8
CLKDIV0_CLKOFFSET.CLK0OFFSET	0x0	No offset
SIGGEN0 Mode and Bit Length Configuration		
SIGGEN0_CTL0.BITLENGTH	0x20	Do 32 shifts.
SIGGEN0_CTL0.MODE	0x1	Configure the mode to shift right once mode. Generates an interrupt after 32 shifts.
SIGGEN0_CTL0.BRIN	0x1	Reverse the bits to the data transform block to make sure that the MSB is transmitted first.
SIGGEN0_CTL0.BROUT	0x1	Reverse the data back and store in the register
SIGGEN0_DATA0[15:0]	0xAA55	Data to be shifted out
SIGGEN0_DATA0[31:16]	0xCCCC	Data to be shifted out
SIGGEN0_DATA1[15:0]	0xAA55	Data to be shifted out
SIGGEN0_DATA1[31:16]	0x55AA	Data to be shifted out

21.6.4 EPG Example: Clock and Data Pair

Example 21-4 register configuration shifts out a data word, also an associated clock is generated to latch the data (EPGOUT1), the data rate is set to divide by 8 and MSB is shifted out first. After 32 shifts are complete, an interrupt is generated for further sequencing.

Example 21-4. Clock and Data Pair Register Configuration

Register	Value	Selected Mode
Epg1MuxRegs		
EPGMXSEL0.SEL0	0x1	Select EPGOUT0 to drive DATAOUT[0]
EPGMXSEL0.SEL1	0x1	Select EPGOUT1 to drive DATAOUT[1]
Epg1Regs		
Global Settings		
GCTL0.EPGOUT0SEL	0x0	Selects signal mux output on EPGOUT0
GCTL3.EPGOUT0_SIGOUTSEL	0x4	Select SIGGEN0.OUT[4] on EPGOUT0, on 64-bit reversal, 31 bit appears on 32 bit (hence, configuring to 4).
GCTL0.EPGOUT1SEL	0x0	Selects signal mux output on EPGOUT1
GCTL3.EPGOUT1_SIGOUTSEL	0x8	Select SIGGEN1.OUT[0] on EPGOUT1.
GCTL1.SIGGEN0_CLKSEL	0x0	Select CLKOUT0 of CLKGEN0 as the clock source of SIGGEN0
GCTL1.SIGGEN1_CLKSEL	0x4	Select CLKOUT0 of CLKGEN1 as the clock source of SIGGEN1
CLKGEN0 Setting		
CLKDIV0_CTL0.PRD	0x7	Divide by 8
CLKDIV0_CLKOFFSET.CLK0OFFSET	0x0	No offset
CLKDIV1_CTL0.PRD	0x3	Divide by 4
SIGGEN0 Mode and Bit Length Configuration		
SIGGEN0_CTL0.BITLENGTH	0x20	Do 32 shifts.
SIGGEN0_CTL0.MODE	0x1	Configure the mode to shift right once mode. Generates an interrupt after 32 shifts.
SIGGEN0_CTL0.BRIN	0x1	Reverse the bits to the data transform block to make sure that the MSB is transmitted first.
SIGGEN0_CTL0.BROUT	0x1	Reverse the data back and store in the register
SIGGEN0_DATA0[15:0]	0xAA55	Data to be shifted out
SIGGEN0_DATA0[31:16]	0xCCCC	Data to be shifted out
SIGGEN0_DATA1[15:0]	0xAA55	Data to be shifted out
SIGGEN0_DATA1[31:16]	0x55AA	Data to be shifted out
SIGGEN1 Mode and Bit Length Configuration		
SIGGEN1_CTL0.BITLENGTH	0x2	Set bit length to 2 to generate a 50% duty clock.
SIGGEN1_CTL0.MODE	0x3	Configure the mode to rotate right repeat mode. Generates a 50% duty cycle clock.
SIGGEN1_DATA0[15:0]	0x2	Data to be shifted out

21.6.5 EPG Example: Clock and Skewed Data Pair

Example 21-5 register configuration shows the data shifted out is skewed by few a EPG input clock cycles.

Example 21-5. Clock and Skewed Data Pair Register Configuration

Register	Value	Selected Mode
Epg1MuxRegs		
EPGMXSEL0.SEL0	0x1	Select EPGOUT0 to drive DATAOUT[0]
EPGMXSEL0.SEL1	0x1	Select EPGOUT1 to drive DATAOUT[1]
Epg1Regs		
Global Settings		
GCTL0.EPGOUT0SEL	0x0	Selects signal mux output on EPGOUT0
GCTL3.EPGOUT0_SIGOUTSEL	0x4	Select SIGGEN0.OUT[4] on EPGOUT0, on 64-bit reversal, 31 bit appears on 32 bit (hence, configuring to 4).
GCTL0.EPGOUT1SEL	0x0	Selects signal mux output on EPGOUT1
GCTL3.EPGOUT1_SIGOUTSEL	0x8	Select SIGGEN1.OUT[0] on EPGOUT1.
GCTL1.SIGGEN0_CLKSEL	0x0	Select CLKOUT0 of CLKGEN0 as the clock source of SIGGEN0
GCTL1.SIGGEN1_CLKSEL	0x4	Select CLKOUT0 of CLKGEN1 as the clock source of SIGGEN1
CLKGEN0 Setting		
CLKDIV0_CTL0.PRD	0x7	Divide by 8
CLKDIV0_CLKOFFSET.CLK0OFFSET	0x2	Offset of 2 cycles.
CLKDIV1_CTL0.PRD	0x3	Divide by 4
SIGGEN0 Mode and Bit Length Configuration		
SIGGEN0_CTL0.BITLENGTH	0x20	Do 32 shifts.
SIGGEN0_CTL0.MODE	0x1	Configure the mode to shift right once mode. Generates an interrupt after 32 shifts.
SIGGEN0_CTL0.BRIN	0x1	Reverse the bits to the data transform block to make sure that the MSB is transmitted first.
SIGGEN0_CTL0.BROUT	0x1	Reverse the data back and store in the register
SIGGEN0_DATA0[15:0]	0xAA55	Data to be shifted out
SIGGEN0_DATA0[31:16]	0xCCCC	Data to be shifted out
SIGGEN0_DATA1[15:0]	0xAA55	Data to be shifted out
SIGGEN0_DATA1[31:16]	0x55AA	Data to be shifted out
SIGGEN1 Mode and Bit Length Configuration		
SIGGEN1_DATA0[15:0]	0x2	Data to be shifted out
SIGGEN1_CTL0.BITLENGTH	0x2	Set bit length to 2 to generate a 50% duty clock.
SIGGEN1_CTL0.MODE	0x3	Configure the mode to rotate right repeat mode. Generates a 50% duty cycle clock.

21.6.6 EPG Example: Capturing Serial Data with a Known Baud Rate

Example 21-6 register configuration captures a 32-bit data stream. The data is generated from SIGGEN0 and captured in SIGGEN1 (EPGOUT0 looped back as EPGIN0). The clock to SIGEN1 is offset by a few cycles to reliably capture the data.

Example 21-6. Capturing Serial Data with a Known Baud Rate Register Configuration

Register	Value	Selected Mode
Epg1MuxRegs		
EPGMXSEL0.SEL0	0x1	Select EPGOUT0 to drive DATAOUT[0]
EPGMXSEL0.SEL1	0x1	Select EPGOUT1 to drive DATAOUT[1]
Epg1Regs		
Global Settings		
GCTL0.EPGOUT0SEL	0x0	Selects signal mux output on EPGOUT0
GCTL3.EPGOUT0_SIGOUTSEL	0x4	Select SIGGEN0.OUT[4] on EPGOUT0, on 64-bit reversal, 31 bit appears on 32 bit (hence, configuring to 4).
GCTL0.EPGOUT1SEL	0x0	Selects signal mux output on EPGOUT1
GCTL3.EPGOUT1_SIGOUTSEL	0x8	Select SIGGEN1.OUT[0] on EPGOUT1.
GCTL1.SIGGEN0_CLKSEL	0x0	Select CLKOUT0 of CLKGEN0 as the clock source of SIGGEN0
GCTL1.SIGGEN1_CLKSEL	0x4	Select CLKOUT0 of CLKGEN1 as the clock source of SIGGEN1
CLKGEN0 Setting		
CLKDIV0_CTL0.PRD	0x7	Divide by 8
CLKDIV0_CLKOFFSET.CLK0OFFSET	0x0	No offset
CLKDIV1_CTL0.PRD	0x7	Divide by 8
CLKDIV1_CLKOFFSET.CLK0OFFSET	0x2	Offset the capture clock by a few cycles to capture the data reliably.
SIGGEN0 Mode and Bit Length Configuration		
SIGGEN0_CTL0.BITLENGTH	0x20	Do 32 shifts, data out looped back to EPGIN[0] to demonstrate the data capture function.
SIGGEN0_CTL0.MODE	0x1	Configure the mode to shift right once mode. Generates an interrupt after 32 shifts.
SIGGEN0_CTL0.BRIN	0x0	Reverse the bits to the data transform block to make sure that the MSB is transmitted first.
SIGGEN0_CTL0.BROUT	0x0	Reverse the data back and store in the register
SIGGEN_DATA0[15:0]	0xAA55	Data to be shifted out
SIGGEN_DATA0[31:16]	0xCCCC	Data to be shifted out
SIGGEN_DATA1[15:0]	0xAA55	Data to be shifted out
SIGGEN_DATA1[31:16]	0x55AA	Data to be shifted out
SIGGEN1 Mode and Bit Length Configuration		
SIGGEN1_CTL0.BITLENGTH	0x32	Set bit length to 2 to generate a 50% duty clock.
SIGGEN1_CTL0.MODE	0x4	Configure the mode to shift left once mode. To capture the data pattern on EPGIN[0].
SIGGEN1_CTL1.DATA0_INSEL	0x1	Select EPGIN[0] as the data input of bit 0.
SIGGEN1_DATA0	0x0	Clear the data contents. Holds 0xAA55 3333 at the end of 32 shifts.

21.7 EPG Interrupt

EPG interrupt can be generated on “BIT_LENGTH" shifts or rotates, or “BIT_LENGTH/2” shifts or rotates. Interrupts can be generated in any of the signal generator modes.

The GINTSTS, GINTEN, GINTCLR, and GINTFRC registers are used to configure the EPG interrupt, as shown in Figure 21-7.

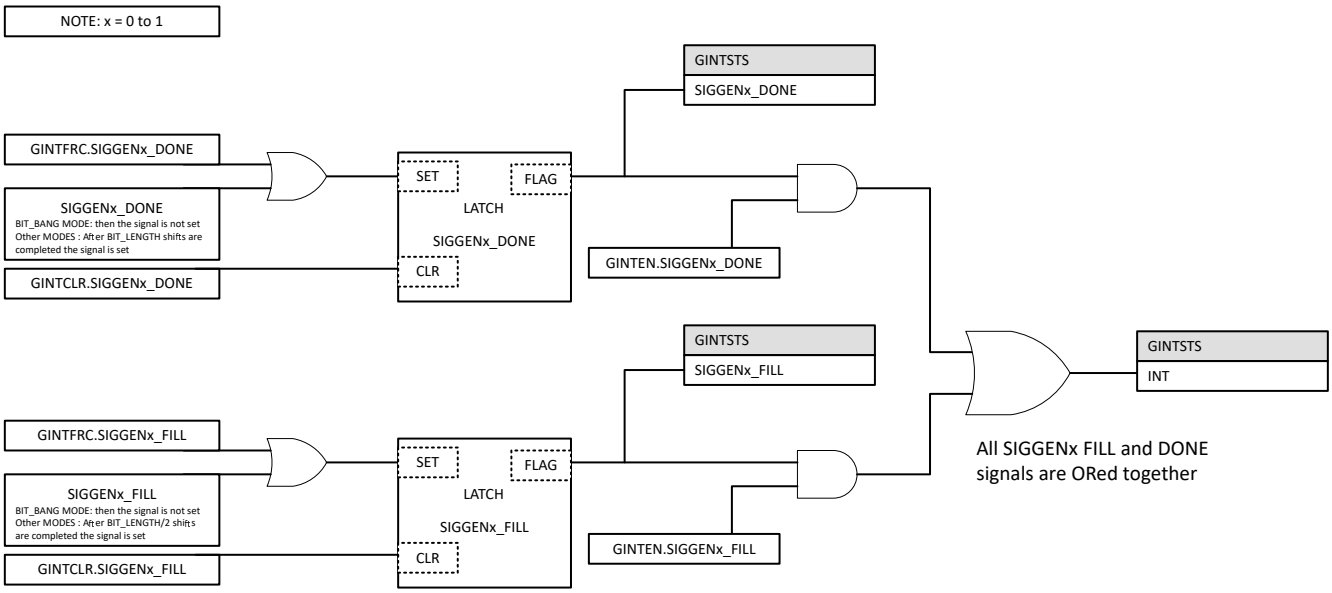


Figure 21-7. EPG Interrupt

21.8 Software

21.8.1 EPG Registers to Driverlib Functions

Table 21-5. EPG Registers to Driverlib Functions

File	Driverlib Function
GCTL0	
epg.h	EPG_enableGlobal
epg.h	EPG_disableGlobal
epg.h	EPG_selectEPGOutput
epg.h	EPG_enableSignalGen
epg.h	EPG_disableSignalGen
GCTL1	
epg.h	EPG_selectSigGenClkSource
GCTL2	
epg.h	EPG_selectClkOutput
GCTL3	
epg.h	EPG_selectSignalOutput
LOCK	
epg.h	EPG_lockReg
epg.h	EPG_releaseLockReg
COMMIT	
epg.h	EPG_commitRegLock
GINTSTS	
epg.h	EPG_getInterruptStatus
GINTEN	
epg.h	EPG_enableInterruptFlag
epg.h	EPG_disableInterruptFlag
GINTCLR	
epg.h	EPG_clearInterruptFlag
GINTFRC	
epg.h	EPG_forceInterruptFlag
CLKDIV0_CTL0	
epg.h	EPG_setClkGenPeriod
epg.h	EPG_setClkGenStopEdge
CLKDIV0_CLKOFFSET	
epg.h	EPG_setClkGenOffset
CLKDIV1_CTL0	
-	See CLKDIV0_CTL0
CLKDIV1_CLKOFFSET	
-	See CLKDIV0_CLKOFFSET
SIGGEN0_CTL0	
epg.h	EPG_setSignalGenMode
epg.h	EPG_enableBitRevOnDataIn
epg.h	EPG_disableBitRevOnDataIn
epg.h	EPG_enableBitRevOnDataOut
epg.h	EPG_disableBitRevOnDataOut
epg.h	EPG_setDataBitLen

Table 21-5. EPG Registers to Driverlib Functions (continued)

File	Driverlib Function
SIGGEN0_CTL1	
epg.h	EPG_setData0In
epg.h	EPG_setData63In
SIGGEN0_DATA0	
epg.h	EPG_setData0Word
epg.h	EPG_getData0ActiveReg
SIGGEN0_DATA1	
epg.h	EPG_setData1Word
epg.h	EPG_getData1ActiveReg
SIGGEN0_DATA0_ACTIVE	
epg.h	EPG_getData0ActiveReg
SIGGEN0_DATA1_ACTIVE	
epg.h	EPG_getData1ActiveReg
REVISION	
-	
MXSEL0	
epg.c	EPG_selectEPGDataOut
MXSELLOCK	
epg.h	EPG_lockMXSelReg
epg.h	EPG_releaseLockMXSelReg
MXSELCOMMIT	
epg.h	EPG_commitMXSelRegLock

21.8.2 EPG Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/epg

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

21.8.2.1 EPG Generating Synchronous Clocks - SINGLE_CORE

FILE: epg_ex1_generate_clocks.c

This example shows how to generate 2 synchronous clocks with edges being offset by 2 clock cycles. It configures Signal Generator to shift a periodic data. Generated Clock has period EPG CLOCK/6.

External Connections

- None. Signal is generated on GPIO 34, 3. Can be visualized through oscilloscope.

Watch Variables

- sigGenActiveData - Active Data of signal generator transform output

21.8.2.2 EPG Generating Two Offset Clocks - SINGLE_CORE

FILE: epg_ex2_generate_two_offset_clocks.c

This example generates two offset clocks using the CLKGEN (CLKDIV) modules. For more information on this example, visit: *Designing With the C2000 Embedded Pattern Generator (EPG)*

External Connections

- None. Signal is generated on GPIO 34, 3. Can be visualized through oscilloscope.

21.8.2.3 EPG Generating Two Offset Clocks With SIGGEN - SINGLE_CORE

FILE: epg_ex3_generate_two_offset_clocks_with_siggen.c

This example generates two offset clocks using the SIGGEN module. For more information on this example, visit: *Designing With the C2000 Embedded Pattern Generator (EPG)*

External Connections

- None. Signal is generated on GPIO 34, 3. Can be visualized through oscilloscope.

21.8.2.4 EPG Generate Serial Data - SINGLE_CORE

FILE: epg_ex4_generate_serial_data.c

This example generates SPICLK and SPI DATA signals using the SIGGEN module. For more information on this example, visit: *Designing With the C2000 Embedded Pattern Generator (EPG)*

External Connections

- None. Signal is generated on GPIO 34, 3. Can be visualized through oscilloscope.

21.8.2.5 EPG Generate Serial Data Shift Mode - SINGLE_CORE

FILE: epg_ex5_generate_serial_data_shift_mode.c

This example generates SPICLK and SPI DATA signals using the SIGGEN module in SHIFT mode. For more information on this example, visit: *Designing With the C2000 Embedded Pattern Generator (EPG)*

External Connections

- None. Signal is generated on GPIO 34, 3. Can be visualized through oscilloscope.

21.9 EPG Registers

This Section describes the EPG Registers.

21.9.1 EPG Base Address Table

Table 21-6. EPG Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
EPG_REGS	EPG_BASE	0x701C_0000	YES	YES	YES	YES	YES	YES	-	YES
EPG_MUX_REGS	EPGMUX_BASE	0x701C_0200	YES	YES	YES	YES	YES	YES	-	YES

21.9.2 EPG_REGS Registers

Table 21-7 lists the memory-mapped registers for the EPG_REGS registers. All register offset addresses not listed in Table 21-7 should be considered as reserved locations and the register contents should not be modified.

Table 21-7. EPG_REGS Registers

Offset	Acronym	Register Name	Protection
0h	GCTL0	EPG Global control register 0	
4h	GCTL1	EPG Global control register 1	
8h	GCTL2	EPG Global control register 2	
Ch	GCTL3	EPG Global control register 3	
10h	EPGLOCK	EPG LOCK Register	
14h	EPGCOMMIT	EPG COMMIT register	
18h	GINTSTS	EPG Global interrupt status register.	
1Ch	GINTEN	EPG Global interrupt enable register.	
20h	GINTCLR	EPG Global interrupt clear register.	
24h	GINTFRC	EPG Global interrupt force register.	
30h	CLKDIV0_CTL0	Clock divider 0's control register 0	
3Ch	CLKDIV0_CLKOFFSET	Clock divider 0's clock offset value	
48h	CLKDIV1_CTL0	Clock divider 1's control register 0	
54h	CLKDIV1_CLKOFFSET	Clock divider 1's clock offset value	
60h	SIGGEN0_CTL0	Signal generator 0's control register 0	
64h	SIGGEN0_CTL1	Signal generator 0's control register 1	
70h	SIGGEN0_DATA0	Signal generator 0's data register 0	
74h	SIGGEN0_DATA1	Signal generator 0's data register 1	
78h	SIGGEN0_DATA0_ACTIVE	Signal generator 0's data active register 0	
7Ch	SIGGEN0_DATA1_ACTIVE	Signal generator 0's data active register 1	
A0h	REVISION	IP Revision tie-off value	

Complex bit access types are encoded to fit into small table cells. Table 21-8 shows the codes that are used for access types in this section.

Table 21-8. EPG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		

Table 21-8. EPG_REGS Access Type Codes (continued)

Access Type	Code	Description
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

21.9.2.1 GCTL0 Register (Offset = 0h) [Reset = 00000000h]

GCTL0 is shown in Figure 21-8 and described in Table 21-9.

Return to the Summary Table.

EPG Global control register 0

Figure 21-8. GCTL0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
EPGOUT23SEL	EPGOUT22SEL	EPGOUT21SEL	EPGOUT20SEL	EPGOUT17SEL	EPGOUT16SEL	EPGOUT15SEL	EPGOUT13SEL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED					SIGGEN1_EN	SIGGEN0_EN	EN
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 21-9. GCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	EPGOUT23SEL	R/W	0h	0 : Selects signal mux output 1 : Selects clock mux output Reset type: SYSRSn
14	EPGOUT22SEL	R/W	0h	0 : Selects signal mux output 1 : Selects clock mux output Reset type: SYSRSn
13	EPGOUT21SEL	R/W	0h	0 : Selects signal mux output 1 : Selects clock mux output Reset type: SYSRSn
12	EPGOUT20SEL	R/W	0h	0 : Selects signal mux output 1 : Selects clock mux output Reset type: SYSRSn
11	EPGOUT17SEL	R/W	0h	0 : Selects signal mux output 1 : Selects clock mux output Reset type: SYSRSn
10	EPGOUT16SEL	R/W	0h	0 : Selects signal mux output 1 : Selects clock mux output Reset type: SYSRSn
9	EPGOUT15SEL	R/W	0h	0 : Selects signal mux output 1 : Selects clock mux output Reset type: SYSRSn
8	EPGOUT13SEL	R/W	0h	0 : Selects signal mux output 1 : Selects clock mux output Reset type: SYSRSn
7-3	RESERVED	R	0h	Reserved
2	SIGGEN1_EN	R/W	0h	0 : Signal generator 1 is disabled. 1 : Signal generator 1 is enabled, signal generator functions as per the mode definition. Reset type: SYSRSn

Table 21-9. GCTL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	SIGGEN0_EN	R/W	0h	0 : Signal generator 0 is disabled. 1 : Signal generator 0 is enabled, signal generator functions as per the mode definition. Reset type: SYSRSn
0	EN	R/W	0h	0 : EPG module is disabled 1 : EPG module is enabled, clock generators and signal generators are functional. Reset type: SYSRSn

21.9.2.2 GCTL1 Register (Offset = 4h) [Reset = 00000000h]

GCTL1 is shown in Figure 21-9 and described in Table 21-10.

Return to the Summary Table.

EPG Global control register 1

Figure 21-9. GCTL1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED			RESERVED	SIGGEN0_CLKSEL		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		

Table 21-10. GCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6-4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2-0	SIGGEN0_CLKSEL	R/W	0h	Clock source select of SIGGEN0: 0x0 : CLKGEN0.CLKOUT0_GCLK 0x1 : CLKGEN0.CLKOUT1_GCLK 0x2 : CLKGEN0.CLKOUT2_GCLK 0x3 : CLKGEN0.CLKOUT3_GCLK 0x4 : CLKGEN1.CLKOUT0_GCLK 0x5 : CLKGEN1.CLKOUT1_GCLK 0x6 : CLKGEN1.CLKOUT2_GCLK 0x7 : CLKGEN1.CLKOUT3_GCLK Reset type: SYSRSn

21.9.2.3 GCTL2 Register (Offset = 8h) [Reset = 00000000h]

GCTL2 is shown in Figure 21-10 and described in Table 21-11.

Return to the Summary Table.

EPG Global control register 2

Figure 21-10. GCTL2 Register

31	30	29	28	27	26	25	24
RESERVED	EPGOUT7_CLKOUTSEL			RESERVED	EPGOUT6_CLKOUTSEL		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	EPGOUT5_CLKOUTSEL			RESERVED	EPGOUT4_CLKOUTSEL		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		
15	14	13	12	11	10	9	8
RESERVED	EPGOUT3_CLKOUTSEL			RESERVED	EPGOUT2_CLKOUTSEL		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	EPGOUT1_CLKOUTSEL			RESERVED	EPGOUT0_CLKOUTSEL		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		

Table 21-11. GCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30-28	EPGOUT7_CLKOUTSEL	R/W	0h	Output 7 signal source select: 0x0 : CLKGEN0.CLKOUT0_DCLK 0x1 : CLKGEN0.CLKOUT1_DCLK 0x2 : CLKGEN0.CLKOUT2_DCLK 0x3 : CLKGEN0.CLKOUT3_DCLK 0x4 : CLKGEN1.CLKOUT0_DCLK 0x5 : CLKGEN1.CLKOUT1_DCLK 0x6 : CLKGEN1.CLKOUT2_DCLK 0x7 : CLKGEN1.CLKOUT3_DCLK Reset type: SYSRSn
27	RESERVED	R/W	0h	Reserved
26-24	EPGOUT6_CLKOUTSEL	R/W	0h	Output 6 signal source select: 0x0 : CLKGEN0.CLKOUT0_DCLK 0x1 : CLKGEN0.CLKOUT1_DCLK 0x2 : CLKGEN0.CLKOUT2_DCLK 0x3 : CLKGEN0.CLKOUT3_DCLK 0x4 : CLKGEN1.CLKOUT0_DCLK 0x5 : CLKGEN1.CLKOUT1_DCLK 0x6 : CLKGEN1.CLKOUT2_DCLK 0x7 : CLKGEN1.CLKOUT3_DCLK Reset type: SYSRSn
23	RESERVED	R/W	0h	Reserved
22-20	EPGOUT5_CLKOUTSEL	R/W	0h	Output 5 signal source select: 0x0 : CLKGEN0.CLKOUT0_DCLK 0x1 : CLKGEN0.CLKOUT1_DCLK 0x2 : CLKGEN0.CLKOUT2_DCLK 0x3 : CLKGEN0.CLKOUT3_DCLK 0x4 : CLKGEN1.CLKOUT0_DCLK 0x5 : CLKGEN1.CLKOUT1_DCLK 0x6 : CLKGEN1.CLKOUT2_DCLK 0x7 : CLKGEN1.CLKOUT3_DCLK Reset type: SYSRSn

Table 21-11. GCTL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	RESERVED	R/W	0h	Reserved
18-16	EPGOUT4_CLKOUTSEL	R/W	0h	Output 4 signal source select: 0x0 : CLKGEN0.CLKOUT0_DCLK 0x1 : CLKGEN0.CLKOUT1_DCLK 0x2 : CLKGEN0.CLKOUT2_DCLK 0x3 : CLKGEN0.CLKOUT3_DCLK 0x4 : CLKGEN1.CLKOUT0_DCLK 0x5 : CLKGEN1.CLKOUT1_DCLK 0x6 : CLKGEN1.CLKOUT2_DCLK 0x7 : CLKGEN1.CLKOUT3_DCLK Reset type: SYSRSn
15	RESERVED	R/W	0h	Reserved
14-12	EPGOUT3_CLKOUTSEL	R/W	0h	Output 3 signal source select: 0x0 : CLKGEN0.CLKOUT0_DCLK 0x1 : CLKGEN0.CLKOUT1_DCLK 0x2 : CLKGEN0.CLKOUT2_DCLK 0x3 : CLKGEN0.CLKOUT3_DCLK 0x4 : CLKGEN1.CLKOUT0_DCLK 0x5 : CLKGEN1.CLKOUT1_DCLK 0x6 : CLKGEN1.CLKOUT2_DCLK 0x7 : CLKGEN1.CLKOUT3_DCLK Reset type: SYSRSn
11	RESERVED	R/W	0h	Reserved
10-8	EPGOUT2_CLKOUTSEL	R/W	0h	Output 2 signal source select: 0x0 : CLKGEN0.CLKOUT0_DCLK 0x1 : CLKGEN0.CLKOUT1_DCLK 0x2 : CLKGEN0.CLKOUT2_DCLK 0x3 : CLKGEN0.CLKOUT3_DCLK 0x4 : CLKGEN1.CLKOUT0_DCLK 0x5 : CLKGEN1.CLKOUT1_DCLK 0x6 : CLKGEN1.CLKOUT2_DCLK 0x7 : CLKGEN1.CLKOUT3_DCLK Reset type: SYSRSn
7	RESERVED	R/W	0h	Reserved
6-4	EPGOUT1_CLKOUTSEL	R/W	0h	Output 1 signal source select: 0x0 : CLKGEN0.CLKOUT0_DCLK 0x1 : CLKGEN0.CLKOUT1_DCLK 0x2 : CLKGEN0.CLKOUT2_DCLK 0x3 : CLKGEN0.CLKOUT3_DCLK 0x4 : CLKGEN1.CLKOUT0_DCLK 0x5 : CLKGEN1.CLKOUT1_DCLK 0x6 : CLKGEN1.CLKOUT2_DCLK 0x7 : CLKGEN1.CLKOUT3_DCLK Reset type: SYSRSn
3	RESERVED	R/W	0h	Reserved
2-0	EPGOUT0_CLKOUTSEL	R/W	0h	Output 0 signal source select: 0x0 : CLKGEN0.CLKOUT0_DCLK 0x1 : CLKGEN0.CLKOUT1_DCLK 0x2 : CLKGEN0.CLKOUT2_DCLK 0x3 : CLKGEN0.CLKOUT3_DCLK 0x4 : CLKGEN1.CLKOUT0_DCLK 0x5 : CLKGEN1.CLKOUT1_DCLK 0x6 : CLKGEN1.CLKOUT2_DCLK 0x7 : CLKGEN1.CLKOUT3_DCLK Reset type: SYSRSn

21.9.2.4 GCTL3 Register (Offset = Ch) [Reset = 00000000h]

GCTL3 is shown in Figure 21-11 and described in Table 21-12.

Return to the Summary Table.

EPG Global control register 3

Figure 21-11. GCTL3 Register

31	30	29	28	27	26	25	24
EPGOUT7_SIGOUTSEL				EPGOUT6_SIGOUTSEL			
R/W-0h				R/W-0h			
23	22	21	20	19	18	17	16
EPGOUT5_SIGOUTSEL				EPGOUT4_SIGOUTSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
EPGOUT3_SIGOUTSEL				EPGOUT2_SIGOUTSEL			
R/W-0h				R/W-0h			
7	6	5	4	3	2	1	0
EPGOUT1_SIGOUTSEL				EPGOUT0_SIGOUTSEL			
R/W-0h				R/W-0h			

Table 21-12. GCTL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EPGOUT7_SIGOUTSEL	R/W	0h	Output 7 source select: 0x0 : SIGGEN0.DATATRANOUT0 0x1 : SIGGEN0.DATATRANOUT1 0x2 : SIGGEN0.DATATRANOUT2 0x3 : SIGGEN0.DATATRANOUT3 0x4 : SIGGEN0.DATATRANOUT4 0x5 : SIGGEN0.DATATRANOUT5 0x6 : SIGGEN0.DATATRANOUT6 0x7 : SIGGEN0.DATATRANOUT7 0x8 : SIGGEN1.DATATRANOUT0 0x9 : SIGGEN1.DATATRANOUT1 0xA : SIGGEN1.DATATRANOUT2 0xB : SIGGEN1.DATATRANOUT3 0xC : SIGGEN1.DATATRANOUT4 0xD : SIGGEN1.DATATRANOUT5 0xE : SIGGEN1.DATATRANOUT6 0xF : SIGGEN1.DATATRANOUT7 Reset type: SYSRSn

Table 21-12. GCTL3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27-24	EPGOUT6_SIGOUTSEL	R/W	0h	Output 6 source select: 0x0 : SIGGEN0.DATATRANOUT0 0x1 : SIGGEN0.DATATRANOUT1 0x2 : SIGGEN0.DATATRANOUT2 0x3 : SIGGEN0.DATATRANOUT3 0x4 : SIGGEN0.DATATRANOUT4 0x5 : SIGGEN0.DATATRANOUT5 0x6 : SIGGEN0.DATATRANOUT6 0x7 : SIGGEN0.DATATRANOUT7 0x8 : SIGGEN1.DATATRANOUT0 0x9 : SIGGEN1.DATATRANOUT1 0xA : SIGGEN1.DATATRANOUT2 0xB : SIGGEN1.DATATRANOUT3 0xC : SIGGEN1.DATATRANOUT4 0xD : SIGGEN1.DATATRANOUT5 0xE : SIGGEN1.DATATRANOUT6 0xF : SIGGEN1.DATATRANOUT7 Reset type: SYSRSn
23-20	EPGOUT5_SIGOUTSEL	R/W	0h	Output 5 source select: 0x0 : SIGGEN0.DATATRANOUT0 0x1 : SIGGEN0.DATATRANOUT1 0x2 : SIGGEN0.DATATRANOUT2 0x3 : SIGGEN0.DATATRANOUT3 0x4 : SIGGEN0.DATATRANOUT4 0x5 : SIGGEN0.DATATRANOUT5 0x6 : SIGGEN0.DATATRANOUT6 0x7 : SIGGEN0.DATATRANOUT7 0x8 : SIGGEN1.DATATRANOUT0 0x9 : SIGGEN1.DATATRANOUT1 0xA : SIGGEN1.DATATRANOUT2 0xB : SIGGEN1.DATATRANOUT3 0xC : SIGGEN1.DATATRANOUT4 0xD : SIGGEN1.DATATRANOUT5 0xE : SIGGEN1.DATATRANOUT6 0xF : SIGGEN1.DATATRANOUT7 Reset type: SYSRSn
19-16	EPGOUT4_SIGOUTSEL	R/W	0h	Output 4 source select: 0x0 : SIGGEN0.DATATRANOUT0 0x1 : SIGGEN0.DATATRANOUT1 0x2 : SIGGEN0.DATATRANOUT2 0x3 : SIGGEN0.DATATRANOUT3 0x4 : SIGGEN0.DATATRANOUT4 0x5 : SIGGEN0.DATATRANOUT5 0x6 : SIGGEN0.DATATRANOUT6 0x7 : SIGGEN0.DATATRANOUT7 0x8 : SIGGEN1.DATATRANOUT0 0x9 : SIGGEN1.DATATRANOUT1 0xA : SIGGEN1.DATATRANOUT2 0xB : SIGGEN1.DATATRANOUT3 0xC : SIGGEN1.DATATRANOUT4 0xD : SIGGEN1.DATATRANOUT5 0xE : SIGGEN1.DATATRANOUT6 0xF : SIGGEN1.DATATRANOUT7 Reset type: SYSRSn

Table 21-12. GCTL3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-12	EPGOUT3_SIGOUTSEL	R/W	0h	Output 3 source select: 0x0 : SIGGEN0.DATATRANOUT0 0x1 : SIGGEN0.DATATRANOUT1 0x2 : SIGGEN0.DATATRANOUT2 0x3 : SIGGEN0.DATATRANOUT3 0x4 : SIGGEN0.DATATRANOUT4 0x5 : SIGGEN0.DATATRANOUT5 0x6 : SIGGEN0.DATATRANOUT6 0x7 : SIGGEN0.DATATRANOUT7 0x8 : SIGGEN1.DATATRANOUT0 0x9 : SIGGEN1.DATATRANOUT1 0xA : SIGGEN1.DATATRANOUT2 0xB : SIGGEN1.DATATRANOUT3 0xC : SIGGEN1.DATATRANOUT4 0xD : SIGGEN1.DATATRANOUT5 0xE : SIGGEN1.DATATRANOUT6 0xF : SIGGEN1.DATATRANOUT7 Reset type: SYSRSn
11-8	EPGOUT2_SIGOUTSEL	R/W	0h	Output 2 source select: 0x0 : SIGGEN0.DATATRANOUT0 0x1 : SIGGEN0.DATATRANOUT1 0x2 : SIGGEN0.DATATRANOUT2 0x3 : SIGGEN0.DATATRANOUT3 0x4 : SIGGEN0.DATATRANOUT4 0x5 : SIGGEN0.DATATRANOUT5 0x6 : SIGGEN0.DATATRANOUT6 0x7 : SIGGEN0.DATATRANOUT7 0x8 : SIGGEN1.DATATRANOUT0 0x9 : SIGGEN1.DATATRANOUT1 0xA : SIGGEN1.DATATRANOUT2 0xB : SIGGEN1.DATATRANOUT3 0xC : SIGGEN1.DATATRANOUT4 0xD : SIGGEN1.DATATRANOUT5 0xE : SIGGEN1.DATATRANOUT6 0xF : SIGGEN1.DATATRANOUT7 Reset type: SYSRSn
7-4	EPGOUT1_SIGOUTSEL	R/W	0h	Output 1 source select: 0x0 : SIGGEN0.DATATRANOUT0 0x1 : SIGGEN0.DATATRANOUT1 0x2 : SIGGEN0.DATATRANOUT2 0x3 : SIGGEN0.DATATRANOUT3 0x4 : SIGGEN0.DATATRANOUT4 0x5 : SIGGEN0.DATATRANOUT5 0x6 : SIGGEN0.DATATRANOUT6 0x7 : SIGGEN0.DATATRANOUT7 0x8 : SIGGEN1.DATATRANOUT0 0x9 : SIGGEN1.DATATRANOUT1 0xA : SIGGEN1.DATATRANOUT2 0xB : SIGGEN1.DATATRANOUT3 0xC : SIGGEN1.DATATRANOUT4 0xD : SIGGEN1.DATATRANOUT5 0xE : SIGGEN1.DATATRANOUT6 0xF : SIGGEN1.DATATRANOUT7 Reset type: SYSRSn

Table 21-12. GCTL3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	EPGOUT0_SIGOUTSEL	R/W	0h	Output 0 source select: 0x0 : SIGGEN0.DATATRANOUT0 0x1 : SIGGEN0.DATATRANOUT1 0x2 : SIGGEN0.DATATRANOUT2 0x3 : SIGGEN0.DATATRANOUT3 0x4 : SIGGEN0.DATATRANOUT4 0x5 : SIGGEN0.DATATRANOUT5 0x6 : SIGGEN0.DATATRANOUT6 0x7 : SIGGEN0.DATATRANOUT7 0x8 : SIGGEN1.DATATRANOUT0 0x9 : SIGGEN1.DATATRANOUT1 0xA : SIGGEN1.DATATRANOUT2 0xB : SIGGEN1.DATATRANOUT3 0xC : SIGGEN1.DATATRANOUT4 0xD : SIGGEN1.DATATRANOUT5 0xE : SIGGEN1.DATATRANOUT6 0xF : SIGGEN1.DATATRANOUT7 Reset type: SYSRSn

21.9.2.5 EPGLOCK Register (Offset = 10h) [Reset = 00000000h]

EPGLOCK is shown in Figure 21-12 and described in Table 21-13.

Return to the Summary Table.

EPG LOCK Register

Figure 21-12. EPGLOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						RESERVED	RESERVED
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SIGGEN0_CTL1	SIGGEN0_CTL0	CLKDIV1_CTL0	CLKDIV0_CTL0	GCTL3	GCTL2	GCTL1	GCTL0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 21-13. EPGLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	Reserved
9	RESERVED	R/W	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7	SIGGEN0_CTL1	R/W	0h	0: Writes to SIGGEN0_CTL1 register is allowed. 1: Writes to SIGGEN0_CTL1 register is not allowed. Reset type: SYSRSn
6	SIGGEN0_CTL0	R/W	0h	0: Writes to SIGGEN0_CTL0 register is allowed. 1: Writes to SIGGEN0_CTL0 register is not allowed. Reset type: SYSRSn
5	CLKDIV1_CTL0	R/W	0h	0: Writes to CLKDIV1_CTL0 register is allowed. 1: Writes to CLKDIV1_CTL0 register is not allowed. Reset type: SYSRSn
4	CLKDIV0_CTL0	R/W	0h	0: Writes to CLKDIV0_CTL0 register is allowed. 1: Writes to CLKDIV0_CTL0 register is not allowed. Reset type: SYSRSn
3	GCTL3	R/W	0h	0: Writes to GCTL3 register is allowed. 1: Writes to GCTL3 register is not allowed. Reset type: SYSRSn
2	GCTL2	R/W	0h	0: Writes to GCTL2 register is allowed. 1: Writes to GCTL2 register is not allowed. Reset type: SYSRSn
1	GCTL1	R/W	0h	0: Writes to GCTL1 register is allowed. 1: Writes to GCTL1 register is not allowed. Reset type: SYSRSn
0	GCTL0	R/W	0h	0: Writes to GCTL0 register is allowed. 1: Writes to GCTL0 register is not allowed. Reset type: SYSRSn

21.9.2.6 EPGCOMMIT Register (Offset = 14h) [Reset = 00000000h]

EPGCOMMIT is shown in Figure 21-13 and described in Table 21-14.

Return to the Summary Table.

EPG COMMIT register

Figure 21-13. EPGCOMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						RESERVED	RESERVED
R-0h						R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
SIGGEN0_CTL1	SIGGEN0_CTL0	CLKDIV1_CTL0	CLKDIV0_CTL0	GCTL3	GCTL2	GCTL1	GCTL0
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 21-14. EPGCOMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	Reserved
9	RESERVED	R/WSonce	0h	Reserved
8	RESERVED	R/WSonce	0h	Reserved
7	SIGGEN0_CTL1	R/WSonce	0h	0: Writes to EPGLOCK.SIGGEN0_CTL1 field is allowed. 1: Writes to EPGLOCK.SIGGEN0_CTL1 field is not allowed. Reset type: SYSRSn
6	SIGGEN0_CTL0	R/WSonce	0h	0: Writes to EPGLOCK.SIGGEN0_CTL0 field is allowed. 1: Writes to EPGLOCK.SIGGEN0_CTL0 field is not allowed. Reset type: SYSRSn
5	CLKDIV1_CTL0	R/WSonce	0h	0: Writes to EPGLOCK.CLKDIV1_CTL0 field is allowed. 1: Writes to EPGLOCK.CLKDIV1_CTL0 field is not allowed. Reset type: SYSRSn
4	CLKDIV0_CTL0	R/WSonce	0h	0: Writes to EPGLOCK.CLKDIV0_CTL0 field is allowed. 1: Writes to EPGLOCK.CLKDIV0_CTL0 field is not allowed. Reset type: SYSRSn
3	GCTL3	R/WSonce	0h	0: Writes to EPGLOCK.GCTL3 field is allowed. 1: Writes to EPGLOCK.GCTL3 field is not allowed. Reset type: SYSRSn
2	GCTL2	R/WSonce	0h	0: Writes to EPGLOCK.GCTL2 field is allowed. 1: Writes to EPGLOCK.GCTL2 field is not allowed. Reset type: SYSRSn
1	GCTL1	R/WSonce	0h	0: Writes to EPGLOCK.GCTL1 field is allowed. 1: Writes to EPGLOCK.GCTL1 field is not allowed. Reset type: SYSRSn
0	GCTL0	R/WSonce	0h	0: Writes to EPGLOCK.GCTL0 field is allowed. 1: Writes to EPGLOCK.GCTL0 field is not allowed. Reset type: SYSRSn

21.9.2.7 GINTSTS Register (Offset = 18h) [Reset = 00000000h]

GINTSTS is shown in Figure 21-14 and described in Table 21-15.

Return to the Summary Table.

EPG Global interrupt status register.

Figure 21-14. GINTSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			RESERVED	RESERVED	SIGGEN0_FILL	SIGGEN0_DONE	INT
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h

Table 21-15. GINTSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	RESERVED	R	0h	Reserved
2	SIGGEN0_FILL	R	0h	0: Do not fill data in SIGGEN0 1: Fill data in SIGGEN0 This status bit does not get set in BIT_BANG mode. In all other modes, the SIGGEN0_FILL bit is set high after the signal generator has completed BITLENGTH/2 shifts. Note: For odd values of BITLENGTH, BITLENGTH/2 is rounded down to the nearest integer. Reset type: SYSRSn
1	SIGGEN0_DONE	R	0h	0: Operation of SIGGEN0 is in progress 1: Operation of SIGGEN0 has completed This status bit does not get set in BIT_BANG mode. In all other modes, the SIGGEN0_DONE bit is set high after the signal generator has completed BITLENGTH shifts. Reset type: SYSRSn
0	INT	R	0h	Global interrupt flag. This bit is set when an interrupt is fired, and cleared by writing 1 to GINTCLR.INT. While the INT status bit is set, new EPG interrupts cannot be generated until the bit has been cleared. Reset type: SYSRSn

21.9.2.8 GINTEN Register (Offset = 1Ch) [Reset = 00000000h]

GINTEN is shown in Figure 21-15 and described in Table 21-16.

Return to the Summary Table.

EPG Global interrupt enable register.

Figure 21-15. GINTEN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			RESERVED	RESERVED	SIGGEN0_FILL	SIGGEN0_DONE	RESERVED
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 21-16. GINTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	SIGGEN0_FILL	R/W	0h	0: Disable interrupt generation when SIGGEN0_FILL bits is set. 1: Enable interrupt generation when SIGGEN0_FILL bits is set. Reset type: SYSRSn
1	SIGGEN0_DONE	R/W	0h	0: Disable interrupt generation when SIGGEN0_DONE bits is set. 1: Enable interrupt generation when SIGGEN0_DONE bits is set. Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

21.9.2.9 GINTCLR Register (Offset = 20h) [Reset = 00000000h]

GINTCLR is shown in Figure 21-16 and described in Table 21-17.

Return to the Summary Table.

EPG Global interrupt clear register.

Figure 21-16. GINTCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			RESERVED	RESERVED	SIGGEN0_FILL	SIGGEN0_DONE	INT
R-0h			R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 21-17. GINTCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4	RESERVED	R-0/W1C	0h	Reserved
3	RESERVED	R-0/W1C	0h	Reserved
2	SIGGEN0_FILL	R-0/W1C	0h	0: No effect 1: Clear SIGGEN0_FILL flag bit. Reset type: SYSRSn
1	SIGGEN0_DONE	R-0/W1C	0h	0: No effect 1: Clear SIGGEN0_DONE flag bit. Reset type: SYSRSn
0	INT	R-0/W1C	0h	0: No effect 1: Clear INT flag bit. Reset type: SYSRSn

21.9.2.10 GINTFRC Register (Offset = 24h) [Reset = 00000000h]

GINTFRC is shown in Figure 21-17 and described in Table 21-18.

Return to the Summary Table.

EPG Global interrupt force register.

Figure 21-17. GINTFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			RESERVED	RESERVED	SIGGEN0_FILL	SIGGEN0_DONE	RESERVED
R-0h			R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h

Table 21-18. GINTFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4	RESERVED	R-0/W1S	0h	Reserved
3	RESERVED	R-0/W1S	0h	Reserved
2	SIGGEN0_FILL	R-0/W1S	0h	0: No effect 1: set SIGGEN0_FILL flag bit. Reset type: SYSRSn
1	SIGGEN0_DONE	R-0/W1S	0h	0: No effect 1: set SIGGEN0_DONE flag bit. Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

21.9.2.11 CLKDIV0_CTL0 Register (Offset = 30h) [Reset = 00000000h]

CLKDIV0_CTL0 is shown in Figure 21-18 and described in Table 21-19.

Return to the Summary Table.

Clock divider 0's control register 0

Figure 21-18. CLKDIV0_CTL0 Register

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

Table 21-19. CLKDIV0_CTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18-16	CLKSTOP	R/W	0h	Determines on which of the CLKOUTs edge clock generation is stopped following a clear of SIGGEN0_CTL0.EN. 000 : Stop on CLKOUT0 010 : Stop on CLKOUT1 100 : Stop on CLKOUT2 110 : Stop on CLKOUT3 Reset type: SYSRSn
15-0	PRD	R/W	0h	Clock divider period: Clock divider counter counts up to period (PRD) and snaps back to 0. Reset type: SYSRSn

21.9.2.12 CLKDIV0_CLKOFFSET Register (Offset = 3Ch) [Reset = 00000000h]

CLKDIV0_CLKOFFSET is shown in Figure 21-19 and described in Table 21-20.

Return to the Summary Table.

Clock divider 0's clock offset value

Figure 21-19. CLKDIV0_CLKOFFSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLK3OFFSET								CLK2OFFSET							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLK1OFFSET								CLK0OFFSET							
R/W-0h								R/W-0h							

Table 21-20. CLKDIV0_CLKOFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	CLK3OFFSET	R/W	0h	Number of source clock cycles by which the divided clock output 3 (CLKOUT3) is delayed. Reset type: SYSRSn
23-16	CLK2OFFSET	R/W	0h	Number of source clock cycles by which the divided clock output 2 (CLKOUT2) is delayed. Reset type: SYSRSn
15-8	CLK1OFFSET	R/W	0h	Number of source clock cycles by which the divided clock output 1 (CLKOUT1) is delayed. Reset type: SYSRSn
7-0	CLK0OFFSET	R/W	0h	Number of source clock cycles by which the divided clock output 0 (CLKOUT0) is delayed. Reset type: SYSRSn

21.9.2.13 CLKDIV1_CTL0 Register (Offset = 48h) [Reset = 00000000h]

CLKDIV1_CTL0 is shown in Figure 21-20 and described in Table 21-21.

Return to the Summary Table.

Clock divider 1's control register 0

Figure 21-20. CLKDIV1_CTL0 Register

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

Table 21-21. CLKDIV1_CTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18-16	CLKSTOP	R/W	0h	Determines on which of the CLKOUTs edge clock generation is stopped following a clear of SIGGEN1_CTL0.EN. 000 : Stop on CLKOUT0 010 : Stop on CLKOUT1 100 : Stop on CLKOUT2 110 : Stop on CLKOUT3 Reset type: SYSRSn
15-0	PRD	R/W	0h	Clock divider period: Clock divider counter counts up to period (PRD) and snaps back to 0. Reset type: SYSRSn

21.9.2.14 CLKDIV1_CLKOFFSET Register (Offset = 54h) [Reset = 00000000h]

CLKDIV1_CLKOFFSET is shown in Figure 21-21 and described in Table 21-22.

Return to the Summary Table.

Clock divider 1's clock offset value

Figure 21-21. CLKDIV1_CLKOFFSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLK3OFFSET								CLK2OFFSET							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLK1OFFSET								CLK0OFFSET							
R/W-0h								R/W-0h							

Table 21-22. CLKDIV1_CLKOFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	CLK3OFFSET	R/W	0h	Number of source clock cycles by which the divided clock output 3 (CLKOUT3) is delayed. Reset type: SYSRSn
23-16	CLK2OFFSET	R/W	0h	Number of source clock cycles by which the divided clock output 2 (CLKOUT2) is delayed. Reset type: SYSRSn
15-8	CLK1OFFSET	R/W	0h	Number of source clock cycles by which the divided clock output 1 (CLKOUT1) is delayed. Reset type: SYSRSn
7-0	CLK0OFFSET	R/W	0h	Number of source clock cycles by which the divided clock output 0 (CLKOUT0) is delayed. Reset type: SYSRSn

21.9.2.15 SIGGEN0_CTL0 Register (Offset = 60h) [Reset = 00000000h]

SIGGEN0_CTL0 is shown in Figure 21-22 and described in Table 21-23.

Return to the Summary Table.

Signal generator 0's control register 0

Figure 21-22. SIGGEN0_CTL0 Register

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

Table 21-23. SIGGEN0_CTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	BITLENGTH	R/W	0h	Defines the number bits which participates in the shift rotate operations. Reset type: SYSRSn
15-7	RESERVED	R	0h	Reserved
6	BROUT	R/W	0h	0 : No bit reversal on data output from data transform block 1 : Perform bit reversal on data output from data transform block Reset type: SYSRSn
5	BRIN	R/W	0h	0 : No bit reversal on data input of data transform block 1 : Perform bit reversal on data input of data transform block Reset type: SYSRSn
4	RESERVED	R	0h	Reserved

Table 21-23. SIGGEN0_CTL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	MODE	R/W	0h	0 : BIT_BANG mode, The value written into DATA0 and DATA1 registers appear on the signal generator outputs as is. 1 : SHIFT_RIGHT_ONCE mode, The data value written into (DATA1,DATA0) registers are shifted right by 1 on every clock. Shifting operations stops when BITLENGTH shifts are done and SIGGEN0_CTL0.EN bit is cleared. 2 : ROTATE_RIGHT_ONCE, The data value written into (DATA1,DATA0) registers are rotated right by 1 on every clock. Rotation happens within (DATA1,DATA0)[BITLENGTH-1:0] Rotate operations stops when BITLENGTH shifts are done and SIGGEN0_CTL0.EN bit is cleared. 3 : ROTATE_RIGHT_REPEAT, The data value written into (DATA1,DATA0) registers are rotated right by 1 on every clock. Rotation happens within (DATA1,DATA0)[BITLENGTH-1:0] Rotate operations continue until SIGGEN0_CTL0.EN bit is cleared. 4 : SHIFT_LEFT_ONCE mode, The data value written into (DATA1,DATA0) registers are shifted left by 1 on every clock. Shifting operations stops when BITLENGTH shifts are done and SIGGEN0_CTL0.EN bit is cleared. 5 : ROTATE_LEFT_ONCE, The data value written into (DATA1,DATA0) registers are rotated left by 1 on every clock. Rotation happens within (DATA1,DATA0)[BITLENGTH-1:0] Rotate operations stops when BITLENGTH shifts are done and SIGGEN0_CTL0.EN bit is cleared. 6 : ROTATE_LEFT_REPEAT, The data value written into (DATA1,DATA0) registers are rotated left by 1 on every clock. Rotation happens within (DATA1,DATA0)[BITLENGTH-1:0] Rotate operations continue until SIGGEN0_CTL0.EN bit is cleared. 7 : SHIFT_RIGHT_REPEAT mode, The data value written into (DATA1,DATA0) registers are shifted right by 1 on every clock. Shifting operations stops when SIGGEN0_CTL0.EN bit is cleared. 8 : SHIFT_LEFT_REPEAT mode, The data value written into (DATA1,DATA0) registers are shifted left by 1 on every clock. Shifting operations stops when SIGGEN0_CTL0.EN bit is cleared. Reset type: SYSRSn

21.9.2.16 SIGGEN0_CTL1 Register (Offset = 64h) [Reset = 00000000h]

SIGGEN0_CTL1 is shown in Figure 21-23 and described in Table 21-24.

Return to the Summary Table.

Signal generator 0's control register 1

Figure 21-23. SIGGEN0_CTL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA63_INSEL				RESERVED											
R/W-0h				R-0h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												DATA0_INSEL			
R-0h												R/W-0h			

Table 21-24. SIGGEN0_CTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	DATA63_INSEL	R/W	0h	Source input of bit 63 of Data register. If 0 delects DATA_NEXT[63] else, selects one of the EPGIN inputs. This provides the ability to capture the data. 0x0 : DATA_NEXT[63] 0x1 : EPGIN0 0x2 : EPGIN1 0x3 : EPGIN2 0x4 : EPGIN3 0x5 : EPGIN4 0x6 : EPGIN5 0x7 : EPGIN6 0x8 : EPGIN7 0x9-0xF : 0 Reset type: SYSRSn
27-4	RESERVED	R	0h	Reserved
3-0	DATA0_INSEL	R/W	0h	Source input of bit 0 of Data register. If 0 delects DATA_NEXT[0] else, selects one of the EPGIN inputs. This provides the ability to capture the data. 0x0 : DATA_NEXT[0] 0x1 : EPGIN0 0x2 : EPGIN1 0x3 : EPGIN2 0x4 : EPGIN3 0x5 : EPGIN4 0x6 : EPGIN5 0x7 : EPGIN6 0x8 : EPGIN7 0x9-0xF : 0 Reset type: SYSRSn

21.9.2.17 SIGGEN0_DATA0 Register (Offset = 70h) [Reset = 00000000h]

SIGGEN0_DATA0 is shown in Figure 21-24 and described in Table 21-25.

Return to the Summary Table.

Signal generator 0's data register 0

Figure 21-24. SIGGEN0_DATA0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGGEN_DATA0																															
R/W-0h																															

Table 21-25. SIGGEN0_DATA0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGGEN_DATA0	R/W	0h	Data used in signal bit stream. {SIGGEN_DATA1,SIGGEN_DATA0} together constitutes a 64 bit data stream, which are modified as per the SIGGENx_CTL0.MODE configuration. Reset type: SYSRSn

21.9.2.18 SIGGEN0_DATA1 Register (Offset = 74h) [Reset = 00000000h]

SIGGEN0_DATA1 is shown in Figure 21-25 and described in Table 21-26.

Return to the Summary Table.

Signal generator 0's data register 1

Figure 21-25. SIGGEN0_DATA1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGGEN_DATA1																															
R/W-0h																															

Table 21-26. SIGGEN0_DATA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGGEN_DATA1	R/W	0h	Data used in signal bit stream. {SIGGEN_DATA1,SIGGEN_DATA0} together constitures a 64 bit data stream, which are modified as per the SIGGENx_CTL0.MODE configuration. Reset type: SYSRSn

21.9.2.19 SIGGEN0_DATA0_ACTIVE Register (Offset = 78h) [Reset = 00000000h]

SIGGEN0_DATA0_ACTIVE is shown in Figure 21-26 and described in Table 21-27.

Return to the Summary Table.

Signal generator 0's data active register 0

Figure 21-26. SIGGEN0_DATA0_ACTIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGEN_DATA0																															
R-0h																															

Table 21-27. SIGGEN0_DATA0_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGEN_DATA0	R	0h	This is the lower 32 bits of the 64 bit active register (used in data transformation) Reset type: SYSRSn

21.9.2.20 SIGGEN0_DATA1_ACTIVE Register (Offset = 7Ch) [Reset = 00000000h]

SIGGEN0_DATA1_ACTIVE is shown in Figure 21-27 and described in Table 21-28.

Return to the Summary Table.

Signal generator 0's data active register 1

Figure 21-27. SIGGEN0_DATA1_ACTIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGGEN_DATA1																															
R-0h																															

Table 21-28. SIGGEN0_DATA1_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGGEN_DATA1	R	0h	This is the upper 32 bits of the 64 bit active register (used in data transformation) Reset type: SYSRSn

21.9.2.21 REVISION Register (Offset = A0h) [Reset = 40010801h]

REVISION is shown in Figure 21-28 and described in Table 21-29.

Return to the Summary Table.

IP Revision tie-off value

Figure 21-28. REVISION Register

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

Table 21-29. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	This identifies the scheme revision ID register type implemented for this module Reset type: SYSRSn
29-28	RESERVED	R-0	0h	Reserved
27-16	FUNC	R	1h	Functional Release Number Reflects software-compatability. If there is no software compatability, a unique func number is assigned for compatible modules, the same number is maintained. Reset type: SYSRSn
15-11	RESERVED	R	1h	Reserved
10-8	MAJOR	R	0h	Major Revision Number Represents major changes to the module (e.g. entirely new features are added/changed). The major revision number for this module. Reset type: SYSRSn
7-6	CUSTOM	R	0h	Custom Module Number Indicates a special version of the module. May not be supported by standard software. Reset type: SYSRSn
5-0	MINOR	R	1h	Minor Revision Number Represents minor changes to the module (e.g. enhancements to existing features). The minor revision number for this module. Reset type: SYSRSn

21.9.3 EPG_MUX_REGS Registers

Table 21-30 lists the memory-mapped registers for the EPG_MUX_REGS registers. All register offset addresses not listed in Table 21-30 should be considered as reserved locations and the register contents should not be modified.

Table 21-30. EPG_MUX_REGS Registers

Offset	Acronym	Register Name	Protection
0h	EPGMXSEL0	EPG Mux select register 0	
18h	EPGMXSELLOCK	EPG Mux select register lock	
1Ch	EPGMXSELCOMMIT	EPG Mux select register commit	

Complex bit access types are encoded to fit into small table cells. Table 21-31 shows the codes that are used for access types in this section.

Table 21-31. EPG_MUX_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

21.9.3.1 EPGMXSEL0 Register (Offset = 0h) [Reset = 00000000h]

EPGMXSEL0 is shown in Figure 21-29 and described in Table 21-32.

Return to the Summary Table.

EPG Mux select register 0

Figure 21-29. EPGMXSEL0 Register

31	30	29	28	27	26	25	24
SEL31	SEL30	SEL29	SEL28	SEL27	SEL26	SEL25	SEL24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
SEL23	SEL22	SEL21	SEL20	SEL19	SEL18	SEL17	SEL16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
SEL15	SEL14	SEL13	SEL12	SEL11	SEL10	SEL9	SEL8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 21-32. EPGMXSEL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SEL31	R/W	0h	0: DATAIN[31] is selected 1: EPGOUT[7] is selected Reset type: SYSRSn
30	SEL30	R/W	0h	0: DATAIN[30] is selected 1: EPGOUT[6] is selected Reset type: SYSRSn
29	SEL29	R/W	0h	0: DATAIN[29] is selected 1: EPGOUT[5] is selected Reset type: SYSRSn
28	SEL28	R/W	0h	0: DATAIN[28] is selected 1: EPGOUT[4] is selected Reset type: SYSRSn
27	SEL27	R/W	0h	0: DATAIN[27] is selected 1: EPGOUT[3] is selected Reset type: SYSRSn
26	SEL26	R/W	0h	0: DATAIN[26] is selected 1: EPGOUT[2] is selected Reset type: SYSRSn
25	SEL25	R/W	0h	0: DATAIN[25] is selected 1: EPGOUT[1] is selected Reset type: SYSRSn
24	SEL24	R/W	0h	0: DATAIN[24] is selected 1: EPGOUT[0] is selected Reset type: SYSRSn
23	SEL23	R/W	0h	0: DATAIN[23] is selected 1: EPGOUT[7] is selected Reset type: SYSRSn
22	SEL22	R/W	0h	0: DATAIN[22] is selected 1: EPGOUT[6] is selected Reset type: SYSRSn

Table 21-32. EPGMXSEL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	SEL21	R/W	0h	0: DATAIN[21] is selected 1: EPGOUT[5] is selected Reset type: SYSRSn
20	SEL20	R/W	0h	0: DATAIN[20] is selected 1: EPGOUT[4] is selected Reset type: SYSRSn
19	SEL19	R/W	0h	0: DATAIN[19] is selected 1: EPGOUT[3] is selected Reset type: SYSRSn
18	SEL18	R/W	0h	0: DATAIN[18] is selected 1: EPGOUT[2] is selected Reset type: SYSRSn
17	SEL17	R/W	0h	0: DATAIN[17] is selected 1: EPGOUT[1] is selected Reset type: SYSRSn
16	SEL16	R/W	0h	0: DATAIN[16] is selected 1: EPGOUT[0] is selected Reset type: SYSRSn
15	SEL15	R/W	0h	0: DATAIN[15] is selected 1: EPGOUT[7] is selected Reset type: SYSRSn
14	SEL14	R/W	0h	0: DATAIN[14] is selected 1: EPGOUT[6] is selected Reset type: SYSRSn
13	SEL13	R/W	0h	0: DATAIN[13] is selected 1: EPGOUT[5] is selected Reset type: SYSRSn
12	SEL12	R/W	0h	0: DATAIN[12] is selected 1: EPGOUT[4] is selected Reset type: SYSRSn
11	SEL11	R/W	0h	0: DATAIN[11] is selected 1: EPGOUT[3] is selected Reset type: SYSRSn
10	SEL10	R/W	0h	0: DATAIN[10] is selected 1: EPGOUT[2] is selected Reset type: SYSRSn
9	SEL9	R/W	0h	0: DATAIN[9] is selected 1: EPGOUT[1] is selected Reset type: SYSRSn
8	SEL8	R/W	0h	0: DATAIN[8] is selected 1: EPGOUT[0] is selected Reset type: SYSRSn
7	SEL7	R/W	0h	0: DATAIN[7] is selected 1: EPGOUT[7] is selected Reset type: SYSRSn
6	SEL6	R/W	0h	0: DATAIN[6] is selected 1: EPGOUT[6] is selected Reset type: SYSRSn
5	SEL5	R/W	0h	0: DATAIN[5] is selected 1: EPGOUT[5] is selected Reset type: SYSRSn

Table 21-32. EPGMXSEL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	SEL4	R/W	0h	0: DATAIN[4] is selected 1: EPGOUT[4] is selected Reset type: SYSRSn
3	SEL3	R/W	0h	0: DATAIN[3] is selected 1: EPGOUT[3] is selected Reset type: SYSRSn
2	SEL2	R/W	0h	0: DATAIN[2] is selected 1: EPGOUT[2] is selected Reset type: SYSRSn
1	SEL1	R/W	0h	0: DATAIN[1] is selected 1: EPGOUT[1] is selected Reset type: SYSRSn
0	SEL0	R/W	0h	0: DATAIN[0] is selected 1: EPGOUT[0] is selected Reset type: SYSRSn

21.9.3.2 EPGMXSELLOCK Register (Offset = 18h) [Reset = 00000000h]

EPGMXSELLOCK is shown in Figure 21-30 and described in Table 21-33.

Return to the Summary Table.

EPG Mux select register lock

Figure 21-30. EPGMXSELLOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	EPGMXSEL0
R-0h						R/W-0h	R/W-0h

Table 21-33. EPGMXSELLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	EPGMXSEL0	R/W	0h	0: Writes to EPGMXSEL0 registers are allowed. 1: Writes to EPGMXSEL0 registers are not allowed. Reset type: SYSRSn

21.9.3.3 EPGMXSELCOMMIT Register (Offset = 1Ch) [Reset = 00000000h]

EPGMXSELCOMMIT is shown in Figure 21-31 and described in Table 21-34.

Return to the Summary Table.

EPG Mux select register commit

Figure 21-31. EPGMXSELCOMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	EPGMXSEL0
R-0h						R/WSonce-0h	R/WSonce-0h

Table 21-34. EPGMXSELCOMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	RESERVED	R/WSonce	0h	Reserved
0	EPGMXSEL0	R/WSonce	0h	0: Writes to EPGMXSELLOCK.EPGMXSEL12 field is allowed. 1: Writes to EPGMXSELLOCK.EPGMXSEL12 field is not allowed. Reset type: SYSRSn

Chapter 22

► ANALOG PERIPHERALS

The following chapters describe the analog peripherals.

Technical Reference Manual Overview

The block diagram is shown in Figure 22-1. This Technical Reference Manual is organized into five major sections:

- C29x SYSTEM RESOURCES

  These chapters describe the C29x CPU subsystem, C29x Boot ROM, device configuration, and other system peripherals.

- ANALOG PERIPHERALS

  These chapters describe the general analog subsystem configuration, Analog-to-Digital Converter (ADC), Buffered Digital-to-Analog Converter (DAC), and Comparator Subsystem (CMPSS).

- CONTROL PERIPHERALS

  These chapters describe the Enhanced Capture (eCAP), High-Resolution Capture (HRCAP), Enhanced Pulse-Width Modulator (ePWM) with High-Resolution Pulse-Width Modulator (HRPWM), Enhanced Quadrature Encoder Pulse (eQEP), and Sigma Delta Filter Module (SDFM) peripherals.

- COMMUNICATION PERIPHERALS

  These chapters describe the communication peripherals available to the C29x subsystem such as the EtherCAT, FSI, I2C, PMBUS, UART, LIN, SPI, and SENT.

- SECURITY PERIPHERALS

  This chapter describes the safety peripherals available to the C29x subsystem such as the Hardware Security Module (HSM) and Cryptographic Accelerator.

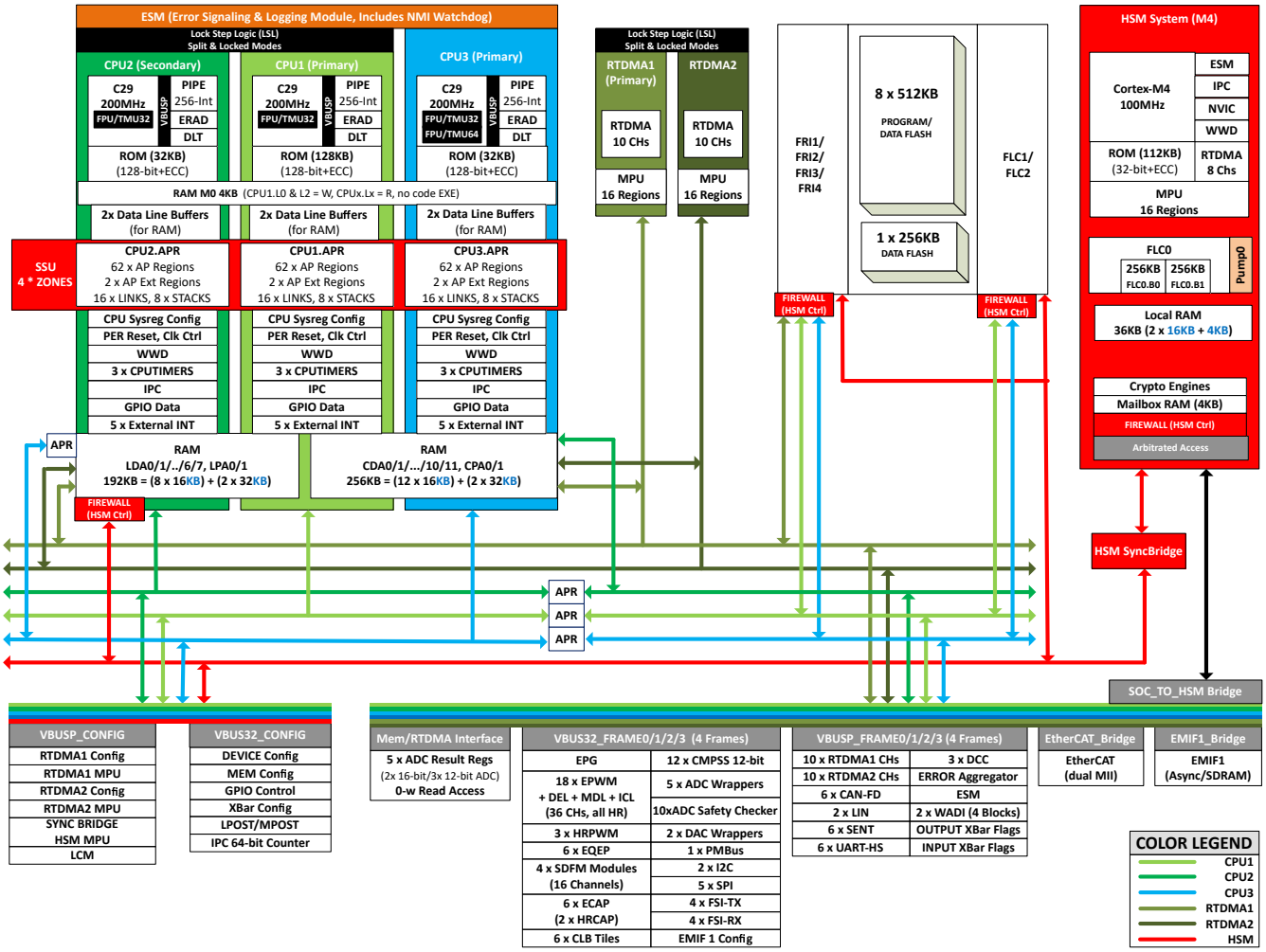


Figure 22-1. Block Diagram

Chapter 23

Analog Subsystem

The analog subsystem module is described in this chapter.

23.1 Introduction

23.1.1 Features

The analog subsystem has the following features:

- Flexible voltage references:
  - The ADCs are referenced to VREFHIx and VREFLOx pins.
    - VREFHIAB and VREFHICDE pin voltages can be driven in externally or can be generated by an internal bandgap voltage reference.
    - The internal voltage reference range can be selected to be 0V to 2.5V for ADC A and ADC B when operated in 16-bit mode; however, the internal voltage reference range can be selected to be 0V to 3.3V or 0V to 2.5V for ADC A and ADC B when operated in 12-bit mode.
    - The internal voltage reference range can be selected to be 0V to 3.3V or 0V to 2.5V for ADC C, ADC D, and ADC E.
  - The buffered DACs are referenced to VREFHIx and VSSA
    - Alternately, these DACs can be referenced to the VDAC pin and VSSA
  - The comparator DACs are referenced to VDDA and VSSA
    - Alternately, these DACs can be referenced to the VDAC pin and VSSA
- Flexible pin usage
  - Buffered DAC outputs, comparator subsystem inputs, and digital inputs (AIOs)/outputs (AGPIOs) are multiplexed with ADC inputs
  - Internal connection to V_{REFLO} for offset self-calibration

23.1.2 Block Diagram

The following analog subsystem block diagrams show the connections between the different integrated analog modules to the device pins. These pins fall into two categories: analog module inputs/outputs and reference pins.

There are two reference pair pins, VREFHIAB /VREFLOAB and VREFHICDE/VREFLOCDE. VREFHIAB and VREFLOAB supply the reference for ADC A and ADC B modules which support both 16-bit and 12-bit mode. VREFHICDE and VREFLOCDE supply ADC C, ADC D and ADC E modules which only support 12-bit mode. VREFHIAB can also be used to supply DAC A, and VREFHICDE can also be used to supply DAC B

The VDAC reference pin can be used to set an alternate range for DAC A and DAC B, and for the DACs inside the CMPSS modules (the CMPSS DACs are referenced to VDDA and VSSA by default). Using this pin as a reference prevents the channel from being used as an ADC input (but the ADC can be used to sample the VDAC voltage, if desired). The choice of reference is configurable per module for each CMPSS or buffered DAC; the selection is made using the module's configuration registers.

Some analog pins support digital functionality through muxed AIOs and AGPIOs. AIOs only support digital input functionality, while AGPIOs support full digital input and output functionality.

The following notes apply to all packages:

- Not all analog pins are available on all devices. See the device data sheet to determine which pins are available.
- See the device data sheet to determine the allowable voltage range for VREFHI and VREFLO.
- An external capacitor is required on the VREFHI pins. See the device data sheet for the specific value required.
- For buffered DAC modules, VSSA is the low reference whether VREFHIx or VDAC is selected as the high reference.
- For CMPSS modules, VSSA is the low reference whether VDAC or VDDA is selected as the high reference.

The following figures show how each analog group is structured. Table 23-3 lists the analog pins and internal connections.

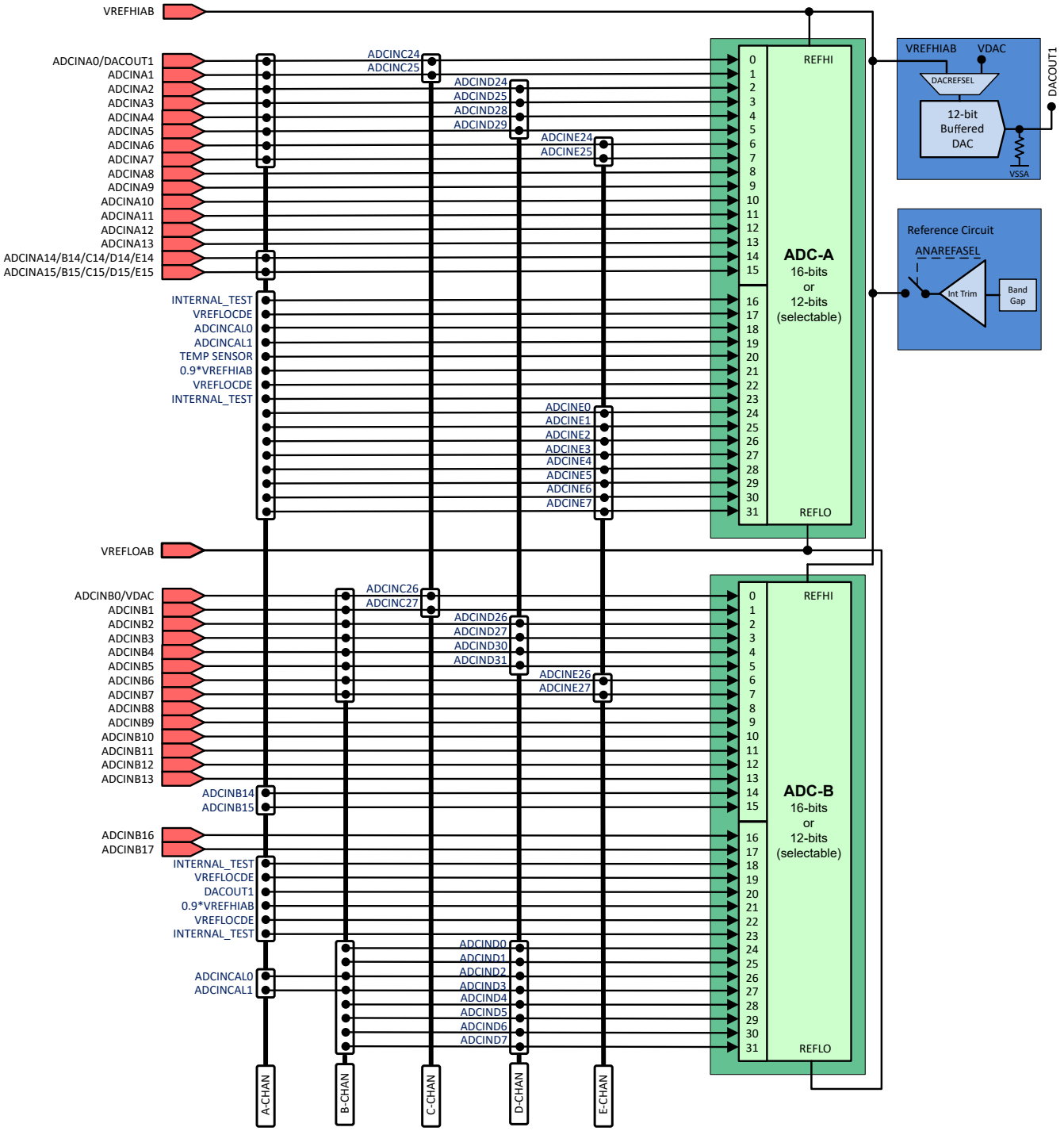


Figure 23-1. Analog Subsystem Block Diagram (ADC A and ADC B)

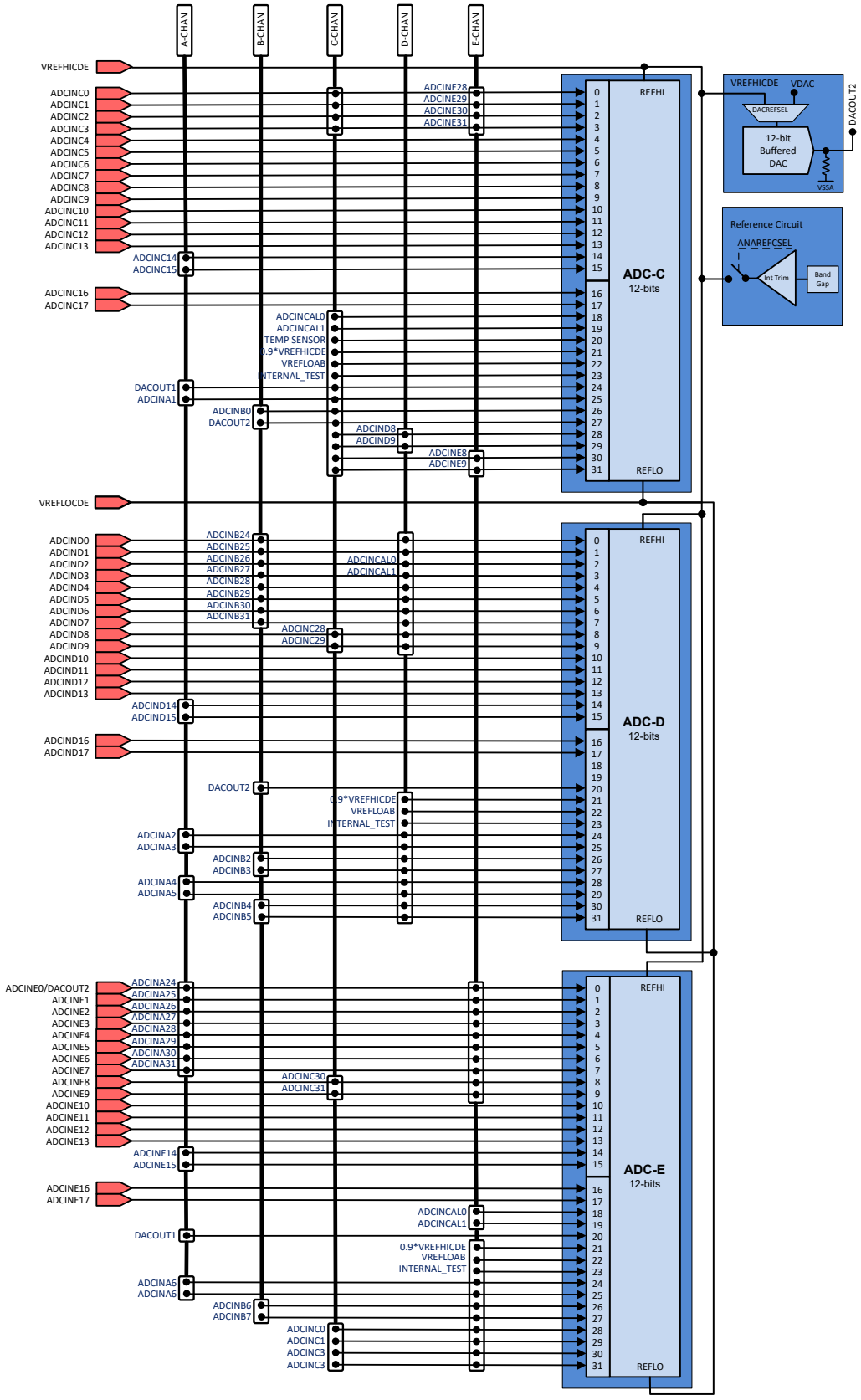


Figure 23-2. Analog Subsystem Block Diagram (ADC C, ADC D, and ADC E)

Input connections to the CMPSS modules are selectable through a programmable input mux. Figure 23-3 shows the CMPSS input connections. Table 23-1 shows the mapping of ADC input signals to CMPSS mux inputs.

- To configure the CMPH_POSIN input mux for CMPSSx, write to the CMPxHPMXSEL field in the CMPHPMXSEL or CMPHPMXSEL1 analog subsystem register.
- To configure the CMPH_NEGIN input mux for CMPSSx, write to the CMPxHNMXSEL field in the CMPHNMXSEL analog subsystem register.
- To configure the CMPL_POSIN input mux for CMPSSx, write to the CMPxLPMXSEL field in the CMPLPMXSEL or CMPLPMXSEL1 analog subsystem register.
- To configure the CMPL_NEGIN input mux for CMPSSx, write to the CMPxLNMXSEL field in the CMPLNMXSEL analog subsystem register.

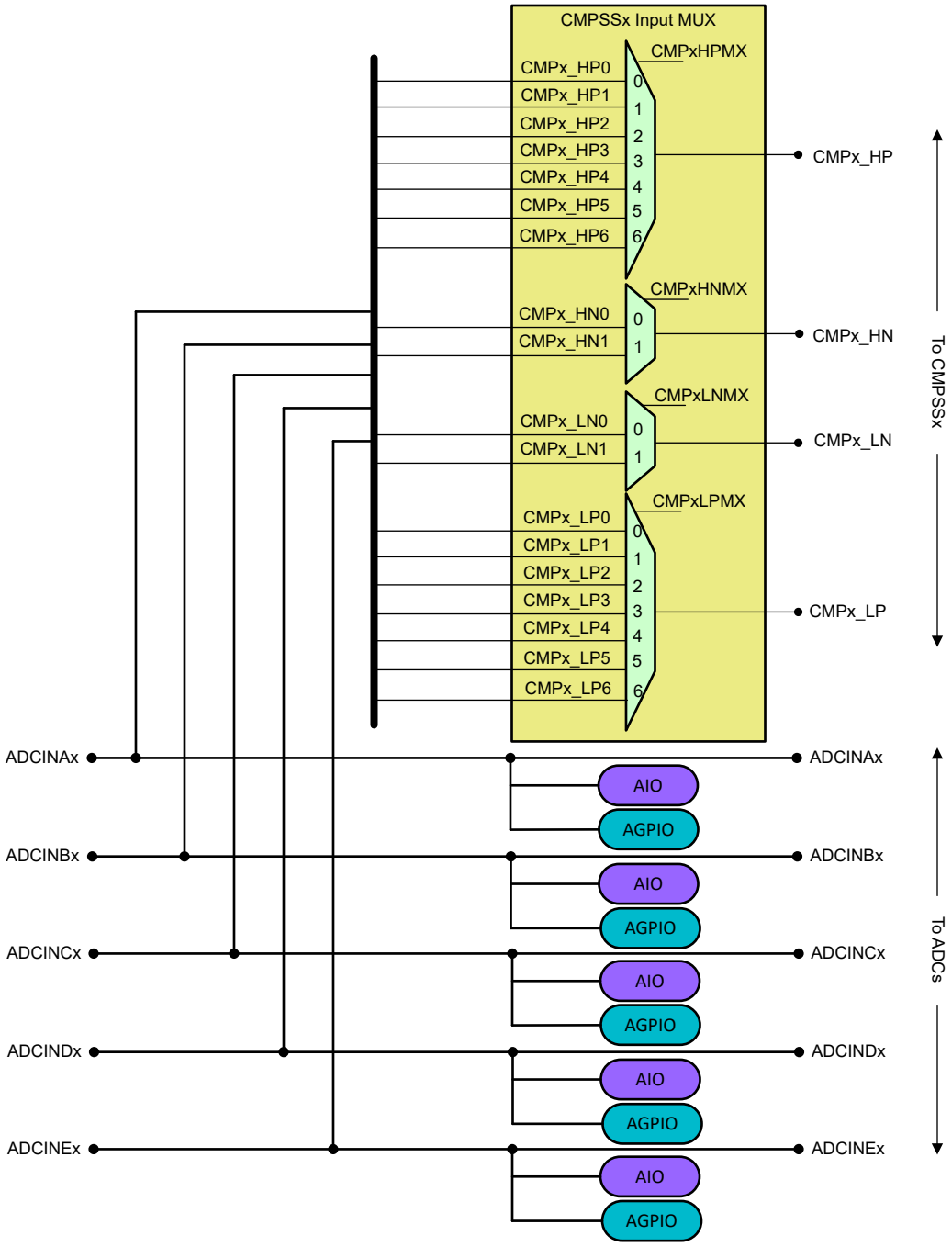


Figure 23-3. Analog Group Connections

Table 23-1. CMPSS Input Mux Options

CMPSSx Input MUX	CMP1	CMP2	CMP3	CMP4	CMP5	CMP6	CMP7	CMP8	CMP9	CMP10	CMP11	CMP12
HP0	A4	A6	B2	A0	D12	D8	D1	D3	C1	C0	C1	C8
HP1	A2	E8	B0	D5	E6	E17	B4	E4	C2	E10	E11	E1
HP2	A3	E9	B1	D0	E7	E16	B5	E5	A7	E12	E13	0.9*VREFHIAB
HP3	B3	D13	TempSense	D2	TempSense	0.9*VREFHIAB	0.9*VREFHICDE	A8	C9	D3	E1	0.9*VREFHICDE
HP4	D6	D7	E2	E3	A8	A9	A10	A11	B6	B7	B8	B9
HP5	A12	A13	A14	A15	C7	C8	C9	C10	B16	B17	C11	C12
HP6	B0	B2	D1	B8	C0	E0	A1	B9	A0	D0	A14	A15
HN0	A5	A7	B3	A1	D13	D9	D2	D4	A2	E8	B6	A6
HN1	A3	A4	B5	D5	E6	E17	B4	E4	E9	D12	C2	B1
LP0	A4	A6	B2	A0	D12	D8	D1	D3	C1	C0	C1	C8
LP1	A2	E8	B0	D5	E6	E17	B4	E4	C2	E10	E11	E1
LP2	A3	E9	B1	D0	E7	E16	B5	E5	A7	E12	E13	0.9*VREFHIAB
LP3	B3	D13	D9	D2	D4	0.9*VREFHIAB	0.9*VREFHICDE	A8	C9	D3	E1	0.9*VREFHICDE
LP4	D6	D7	E2	E3	B10	B11	B12	B13	C3	C4	C5	C6
LP5	A12	A13	A14	A15	C13	C16	C17	D10	D11	D16	D17	E0
LP6	B0	B2	D1	B8	C0	E0	A1	B9	A0	D0	A14	A15
LN0	A5	A7	B3	A1	D13	D9	D2	D4	A2	E8	B6	A6
LN1	A3	A4	B5	D5	E6	E17	B4	E4	E9	D12	C2	B1

23.2 Optimizing Power-Up Time

The analog-to-digital converters (ADC) and buffered digital-to-analog converters (DAC) share a common reference circuit. If needed, an application using one or more of these modules can optimize power-up time by taking advantage of the shared reference. Once one of the modules using the shared reference has been initialized in internal reference mode, the power-up time for subsequent modules can be optimized by subtracting the reference power-up time from the minimum power-up time requirement.

For instance, if ADCA requires $t_{ADCPUINT}$ to power up in internal reference mode, and $t_{ADCPUEXT}$ to power up in external reference mode, the application does not need to wait $t_{ADCPUINT}$ to power up a second ADC instance such as ADCC in internal reference mode. In this case, the application can simply wait for $t_{ADCPUEXT}$ after powering up ADCC, even though both ADCs are used in internal reference mode. In the same scenario, if the application wished to use DACA in internal reference mode, the required wait time after power-up is $t_{DACPUEXT}$, not the longer $t_{DACPUINT}$.

There is also a wait time associated with power-up in internal reference mode when switching between 2.5V and 3.3V range. See the device data sheet for wait time values.

23.3 Digital Inputs on ADC Pins (AIOs)

Some GPIOs are multiplexed with analog pins and only have digital input functionality. These are also referred to as AIOs. Pins with only an AIO option on this port can only function in input mode. See the device data sheet for list of AIO signals. By default, these pins function as analog pins and the GPIOs are in a high-impedance state. The GPyAMSEL register is used to configure these pins for digital or analog operation.

Note

If digital signals with sharp edges (high dv/dt) are connected to the AIOs, cross-talk can occur with adjacent analog signals. Therefore, limit the edge rate of signals connected to AIOs if adjacent channels are being used for analog functions.

23.4 Digital Inputs and Outputs on ADC Pins (AGPIOs)

Some GPIOs are multiplexed with analog pins and have digital input and output functionality. These are also referred to as AGPIOs. Unlike AIOs, AGPIOs have full input and output capability. By default, the AGPIOs are not connected and must be configured. Table 23-2 shows how to configure the AGPIOs. To enable the analog functionality, set the register AGPIOCTRLx from analog subsystem. To enable the digital functionality, set the register GPxAMSEL from the *General-Purpose Input/Output (GPIO)* chapter.

Table 23-2. AGPIO Configuration

AGPIOCTRLx.GPIOy (Default = 0)	GPxAMSEL.GPIOy (Default = 1)	Pin Connected To:	
		ADC	GPIOy
0	0	-	Yes
0	**1**	**-** (1)	**-** (1)
1	0	-	Yes
1	1	Yes	-

(1) By default there are no signals connected to AGPIO pins. One of the other rows in the table must be chosen for pin functionality.

Note

If digital signals with sharp edges (high dv/dt) are connected to the AGPIOs, cross-talk can occur with adjacent analog signals. The user must therefore limit the edge rate of signals connected to AGPIOs, if adjacent channels are being used for analog functions.

23.5 Analog Pins and Internal Connections

Table 23-3. Analog Pin Connections

Pin Name	Pins/Package				ADC					DAC	Comparator Subsystem (Mux)				AIO Input/ GPIO
	256 ZEX	176 PTS	144 RFS	100 PZS	A	B	C	D	E		High Positive	High Negative	Low Positive	Low Negative	
VREFHIAB	M2	38	30	19											
VREFHICDE	R4	54	45	33											
VFEFLOAB	M1	37	29	18			C22	D22	E22						
VREFLOCDE	T4	53	44	32	A17,A22	B19,B22									
Analog Group 1											**CMP1 and other comparators**				
ADCINA3	L4	35	27		A3			D25			CMP1 (HPMXSEL=2)	CMP1 (HNMXSEL=1)	CMP1 (LPMXSEL=2)	CMP1 (LNMXSEL=1)	AIO163
ADCINA5	K5	31	23		A5			D29				CMP1 (HNMXSEL=0)		CMP1 (LNMXSEL=0)	AIO165
ADCINA12	T7				A12						CMP1 (HPMXSEL=5)		CMP1 (LPMXSEL=5)		AIO166
ADCIND6	N10	71	60			B30		D6			CMP1 (HPMXSEL=4)		CMP1 (LPMXSEL=4)		GPIO242
ADCINA4	M7	32	24		A4			D28			CMP1 (HPMXSEL=0)	CMP2 (HNMXSEL=1)	CMP1 (LPMXSEL=0)	CMP2 (LNMXSEL=1)	AIO164
ADCINB0	P2	42	34	23		B0	C26			VDAC	CMP1 (HPMXSEL=6) CMP3 (HPMXSEL=1)		CMP1 (LPMXSEL=6) CMP3 (LPMXSEL=1)		AIO170
ADCINB3	M6	33	25	16		B3		D27			CMP1 (HPMXSEL=3)	CMP3 (HNMXSEL=0)	CMP1 (LPMXSEL=3)	CMP3 (LNMXSEL=0)	AIO173
ADCINA2	N9	36	28		A2			D24			CMP1 (HPMXSEL=1)	CMP9 (HNMXSEL=0)	CMP1 (LPMXSEL=1)	CMP9 (LNMXSEL=0)	AIO162
Analog Group 2											**CMP2 and other comparators**				
ADCINA13	P6				A13						CMP2 (HPMXSEL=5)		CMP2 (LPMXSEL=5)		AIO167
ADCIND7	J3	72	61			B31		D7			CMP2 (HPMXSEL=4)		CMP2 (LPMXSEL=4)		GPIO243
ADCINB2	L5	34	26	17		B2		D26			CMP2 (HPMXSEL=6)		CMP2 (LPMXSEL=6)		AIO172
ADCINB2	L5	34	26	17		B2		D26			CMP3 (HPMXSEL=0)		CMP3 (LPMXSEL=0)		AIO172
ADCIND13	P4							D13			CMP2 (HPMXSEL=3)	CMP5 (HNMXSEL=0)	CMP2 (LPMXSEL=3)	CMP5 (LNMXSEL=0)	AIO199
ADCINA7	G2	25	17	12	A7				E25		CMP9 (HPMXSEL=2)	CMP2 (HNMXSEL=0)	CMP9 (LPMXSEL=2)	CMP2 (LNMXSEL=0)	GPIO225
ADCINE9	N8						C31		E9		CMP2 (HPMXSEL=2)	CMP9 (HNMXSEL=1)	CMP2 (LPMXSEL=2)	CMP9 (LNMXSEL=1)	AIO207
ADCINE8	R8						C30		E8		CMP2 (HPMXSEL=1)	CMP10 (HNMXSEL=0)	CMP2 (LPMXSEL=1)	CMP10 (LNMXSEL=0)	AIO206
ADCINA6	G3	26	18	13	A6				E24		CMP2 (HPMXSEL=0)	CMP12 (HNMXSEL=0)	CMP2 (LPMXSEL=0)	CMP12 (LNMXSEL=0)	GPIO224
Analog Group 3											**CMP3 and other comparators**				
ADCINE2	K1	59	51		A26				E2		CMP3 (HPMXSEL=4)		CMP3 (LPMXSEL=4)		AIO204
TempSensor					A20		C20				CMP3 (HPMXSEL=3) CMP5 (HPMXSEL=3)				
ADCIND9	T10	76					C29	D9				CMP6 (HNMXSEL=0)	CMP3 (LPMXSEL=3)	CMP6 (LNMXSEL=0)	GPIO245
ADCIND1	R3	48	40	29		B25		D1			CMP3 (HPMXSEL=6) CMP7 (HPMXSEL=0)		CMP3 (LPMXSEL=6) CMP7 (LPMXSEL=0)		AIO193
ADCINB5	H3	29	21			B5		D31			CMP7 (HPMXSEL=2)	CMP3 (HNMXSEL=1)	CMP7 (LPMXSEL=2)	CMP3 (LNMXSEL=1)	AIO175
ADCINA14	N4	40	32	21	A14	B14	C14	D14	E14		CMP3 (HPMXSEL=5) CMP11 (HPMXSEL=6)		CMP3 (LPMXSEL=5) CMP11 (LPMXSEL=6)		AIO168
ADCINB1	N3	41	33	22		B1	C27				CMP3 (HPMXSEL=2)	CMP12 (HNMXSEL=1)	CMP3 (LPMXSEL=2)	CMP12 (LNMXSEL=1)	AIO171

Table 23-3. Analog Pin Connections (continued)

Pin Name	Pins/Package				ADC					DAC	Comparator Subsystem (Mux)				AIO Input/ GPIO
	256 ZEX	176 PTS	144 RFS	100 PZS	A	B	C	D	E		High Positive	High Negative	Low Positive	Low Negative	
Analog Group 4											**CMP4 and other comparators**				
ADCIND5	F1	66	55			B29		D5			CMP4 (HPMXSEL=1)	CMP4 (HNMXSEL=1)	CMP4 (LPMXSEL=1)	CMP4 (LNMXSEL=1)	GPIO241
ADCINE3	L2	60	52		A27				E3		CMP4 (HPMXSEL=4)		CMP4 (LPMXSEL=4)		AIO205
ADCINA1	P1	43	35	24	A1		C25				CMP7 (HPMXSEL=6)	CMP4 (HNMXSEL=0)	CMP7 (LPMXSEL=6)	CMP4 (LNMXSEL=0)	AIO161
ADCIND2	J5	57	49	34		B26		D2			CMP4 (HPMXSEL=3)	CMP7 (HNMXSEL=0)	CMP4 (LPMXSEL=3)	CMP7 (LNMXSEL=0)	AIO194
ADCINA0	R1	44	36	25	A0	C24				DACOUT1	CMP4 (HPMXSEL=0)		CMP4 (LPMXSEL=0)		AIO160
											CMP9 (HPMXSEL=6)		CMP9 (LPMXSEL=6)		
ADCIND0	T3	47	39	28		B24		D0			CMP4 (HPMXSEL=2)		CMP4 (LPMXSEL=2)		AIO192
											CMP10 (HPMXSEL=6)		CMP10 (LPMXSEL=6)		
ADCINB8	K4	20	15	11		B8					CMP4 (HPMXSEL=6)		CMP4 (LPMXSEL=6)		GPIO232
											CMP11 (HPMXSEL=4)				
ADCINA15	M4	39	31	20	A15	B15	C15	D15	E15		CMP4 (HPMXSEL=5)		CMP4 (LPMXSEL=5)		AIO169
											CMP12 (HPMXSEL=6)		CMP12 (LPMXSEL=6)		
Analog Group 5											**CMP5 and other comparators**				
ADCINB10	M8	16	13			B10							CMP5 (LPMXSEL=4)		GPIO234
ADCINC7	N11	64					C7				CMP5 (HPMXSEL=5)				GPIO237
ADCINC13	D2						C13						CMP5 (LPMXSEL=5)		AIO189
ADCINE6	T5	73	62		A30				E6		CMP5 (HPMXSEL=1)	CMP5 (HNMXSEL=1)	CMP5 (LPMXSEL=1)	CMP5 (LNMXSEL=1)	GPIO248
ADCINE7	F2	74	63		A31				E7		CMP5 (HPMXSEL=2)		CMP5 (LPMXSEL=2)		GPIO249
ADCINA8	K3	22	16		A8						CMP5 (HPMXSEL=4)				GPIO226
											CMP8 (HPMXSEL=3)		CMP8 (LPMXSEL=3)		
ADCIND4	P11	65				B28		D4				CMP8 (HNMXSEL=0)	CMP5 (LPMXSEL=3)	CMP8 (LNMXSEL=0)	GPIO240
ADCINC0	R2	45	37	26			C0		E28		CMP5 (HPMXSEL=6)		CMP5 (LPMXSEL=6)		AIO180
											CMP10 (HPMXSEL=0)		CMP10 (LPMXSEL=0)		
ADCIND12	R5							D12			CMP5 (HPMXSEL=0)	CMP10 (HNMXSEL=1)	CMP5 (LPMXSEL=0)	CMP10 (LNMXSEL=1)	AIO198
Analog Group 6											**CMP6 and other comparators**				
ADCINA9	J2	21			A9						CMP6 (HPMXSEL=4)				GPIO227
ADCINB11	M9	15	12			B11							CMP6 (LPMXSEL=4)		GPIO235
ADCINC16	L3						C16						CMP6 (LPMXSEL=5)		AIO190
ADCIND8	R9	75					C28	D8			CMP6 (HPMXSEL=0)		CMP6 (LPMXSEL=0)		GPIO244
ADCINE16	J1								E16		CMP6 (HPMXSEL=2)		CMP6 (LPMXSEL=2)		AIO212
ADCINE17	T9								E17		CMP6 (HPMXSEL=1)	CMP6 (HNMXSEL=1)	CMP6 (LPMXSEL=1)	CMP6 (LNMXSEL=1)	AIO213
ADCINC8	R10	69	58	40			C8				CMP6 (HPMXSEL=5)				GPIO238
											CMP12 (HPMXSEL=0)		CMP12 (LPMXSEL=0)		
ADCINE0	P3	49	41	30	A24				E0	DACOUT2	CMP6 (HPMXSEL=6)		CMP6 (LPMXSEL=6)		AIO202
													CMP12 (LPMXSEL=5)		

Table 23-3. Analog Pin Connections (continued)

Pin Name	Pins/Package				ADC					DAC	Comparator Subsystem (Mux)				AIO Input/ GPIO
	256 ZEX	176 PTS	144 RFS	100 PZS	A	B	C	D	E		High Positive	High Negative	Low Positive	Low Negative	
0.9*VREFHIAB					A21	B21					CMP6 (HPMXSEL=3)		CMP6 (LPMXSEL=3)		
											CMP12 (HPMXSEL=2)		CMP12 (LPMXSEL=2)		
Analog Group 7											**CMP7 and other comparators**				
ADCINA10	D1	18			A10						CMP7 (HPMXSEL=4)				GPIO228
ADCINB4	H2	30	22			B4		D30			CMP7 (HPMXSEL=1)	CMP7 (HNMXSEL=1)	CMP7 (LPMXSEL=1)	CMP7 (LNMXSEL=1)	AIO174
ADCINB12	E3					B12							CMP7 (LPMXSEL=4)		AIO176
ADCINC17	K2						C17						CMP7 (LPMXSEL=5)		AIO191
ADCINC9	P10	70	59	41			C9				CMP7 (HPMXSEL=5)				GPIO239
											CMP9 (HPMXSEL=3)		CMP9 (LPMXSEL=3)		
0.9*VREFHICDE							C21	D21	E21		CMP7 (HPMXSEL=3)		CMP7 (LPMXSEL=3)		
											CMP12 (HPMXSEL=3)		CMP12 (LPMXSEL=3)		
Analog Group 8											**CMP8 and other comparators**				
ADCINB13	D3					B13							CMP8 (LPMXSEL=4)		AIO177
ADCINA11	C1	17			A11						CMP8 (HPMXSEL=4)				GPIO229
ADCINC10	C2						C10				CMP8 (HPMXSEL=5)				AIO186
ADCIND10	P9							D10					CMP8 (LPMXSEL=5)		AIO196
ADCINE4	E2	67	56	38	A28				E4		CMP8 (HPMXSEL=1)	CMP8 (HNMXSEL=1)	CMP8 (LPMXSEL=1)	CMP8 (LNMXSEL=1)	GPIO246
ADCINE5	N6	68	57	39	A29				E5		CMP8 (HPMXSEL=2)		CMP8 (LPMXSEL=2)		GPIO247
ADCIND3	L1	58	50	35		B27		D3			CMP8 (HPMXSEL=0)		CMP8 (LPMXSEL=0)		AIO195
											CMP10 (HPMXSEL=3)		CMP10 (LPMXSEL=3)		
ADCINB9	E1	19	14	10		B9					CMP8 (HPMXSEL=6)		CMP8 (LPMXSEL=6)		GPIO233
											CMP12 (HPMXSEL=4)				
Analog Group 9											**CMP9 and other comparators**				
ADCINB16	C3					B16					CMP9 (HPMXSEL=5)				AIO178
ADCINC3	N5	52	44				C3		E30				CMP9 (LPMXSEL=4)		AIO183
ADCIND11	M10							D11					CMP9 (LPMXSEL=5)		AIO197
ADCINB6	G1	24				B6			E26		CMP9 (HPMXSEL=4)	CMP11 (HNMXSEL=0)		CMP11 (LNMXSEL=0)	GPIO230
ADCINC1	T2	46	38	27			C1		E29		CMP9 (HPMXSEL=0)		CMP9 (LPMXSEL=0)		AIO181
											CMP11 (HPMXSEL=0)		CMP11 (LPMXSEL=0)		
ADCINC2	P5	51	43				C2		E30		CMP9 (HPMXSEL=1)		CMP9 (LPMXSEL=1)		AIO182
												CMP11 (HNMXSEL=1)		CMP11 (LNMXSEL=1)	
Analog Group 10											**CMP10 and other comparators**				
ADCINB7	H1	23				B7			E27		CMP10 (HPMXSEL=4)				GPIO231
ADCINB17	F3					B17					CMP10 (HPMXSEL=5)				AIO179
ADCINC4	M5	55	47				C4						CMP10 (LPMXSEL=4)		AIO184
ADCIND16	R7							D16					CMP10 (LPMXSEL=5)		AIO200
ADCINE10	P8								E10		CMP10 (HPMXSEL=1)		CMP10 (LPMXSEL=1)		AIO208

Table 23-3. Analog Pin Connections (continued)

Pin Name	Pins/Package				ADC					DAC	Comparator Subsystem (Mux)				AIO Input/ GPIO
	256 ZEX	176 PTS	144 RFS	100 PZS	A	B	C	D	E		High Positive	High Negative	Low Positive	Low Negative	
ADCINE12	P7								E12		CMP10 (HPMXSEL=2)		CMP10 (LPMXSEL=2)		AIO210
Analog Group 11											**CMP11 and other comparators**				
ADCINC5	J4	56	48				C5						CMP11 (LPMXSEL=4)		AIO185
ADCINC11	B2						C11				CMP11 (HPMXSEL=5)				AIO187
ADCIND17	T8							D17					CMP11 (LPMXSEL=5)		AIO201
ADCINE11	N7								E11		CMP11 (HPMXSEL=1)		CMP11 (LPMXSEL=1)		AIO209
ADCINE13	R6								E13		CMP11 (HPMXSEL=2)		CMP11 (LPMXSEL=2)		AIO211
ADCINE1	M3	50	42	31	A25				E1		CMP11 (HPMXSEL=3)		CMP11 (LPMXSEL=3)		AIO203
											CMP12 (HPMXSEL=1)		CMP12 (LPMXSEL=1)		
Analog Group 12											**CMP12 and other comparators**				
ADCINC6	M11	63					C6						CMP12 (LPMXSEL=4)		GPIO236
ADCINC12	B3						C12				CMP12 (HPMXSEL=5)				AIO188

Table 23-4. Analog Signal Descriptions

Signal Name	Description
ADCINAx, Ax	ADC A Input
ADCINBx, Bx	ADC B Input
ADCINCx, Cx	ADC C Input
ADCINDx, Dx	ADC D Input
ADCINEx, Ex	ADC E Input
CMPx_HNy	Comparator subsystem high comparator negative input
CMPx_HPy	Comparator subsystem high comparator positive input
CMPx_LNy	Comparator subsystem low comparator negative input
CMPx_LPy	Comparator subsystem low comparator positive input
DACOUTx	Buffered DAC Output
TEMP SENSOR, TS	Internal temperature sensor
VDAC	Optional external reference voltage for on-chip DACs. There is a 100pF capacitor to VSSA on this pin whether used for ADC input or DAC reference that cannot be disabled. If this pin is being used as a reference for the on-chip DACs, place at least a 1µF capacitor on this pin.

Table 23-5. Reference Summary

Module	Reference Option	Configured Where?	Register	Driverlib Function	Notes
ADC	External or Internal	Analog Subsystem	AnalogSubsysRegs.ANAREFACTL.bit.ANAREFxSEL AnalogSubsysRegs.ANAREFCCTL.bit.ANAREFxSEL	ASysCtl_setAnalogReferenceInternal, ASysCtl_setAnalogReferenceExternal	Internal reference for ADCA/B and ADCC/D/E.
	Internal Reference 2.5V or 3.3V	Analog Subsystem	AnalogSubsysRegs.ANAREFxCTL.bit.ANAREFx2P5SEL	ASysCtl_setAnalogReference2P5, ASysCtl_setAnalogReference1P65	
Buffered DAC	VREFHI or VDAC	DAC Module	DacxRegs. DACCTL.bit.DACREFSEL	DAC_CTL_DACREFSEL	
	External or Internal	Analog Subsystem	AnalogSubsysRegs.ANAREFACTL.bit.ANAREFxSEL AnalogSubsysRegs.ANAREFCCTL.bit.ANAREFxSEL	ASysCtl_setAnalogReferenceInternal, ASysCtl_setAnalogReferenceExternal	Internal reference for ADCA/B and ADCC/D/E.
CMPSS DACs	VDDA or VDAC	CMPSS Module	CmpssxRegs. COMPDACCTL.bit.SELREF	CMPSS_COMPDACCTL_SELREF	

23.6 Software

23.6.1 ASYSCTL Registers to Driverlib Functions

Table 23-6. ASYSCTL Registers to Driverlib Functions

File	Driverlib Function
PMMVREGTRIM	
-	
CTLTRIMSTS	
-	
REFBUFCONFIGCDE	
-	
INTERNALTESTCTL	
adc.h	ADC_disableIntRefloConnection
asysctl.h	ASysCtl_selectInternalTestNode
CONFIGLOCK	
asysctl.h	ASysCtl_lockAGPIOFilter
asysctl.h	ASysCtl_lockAGPIOCtrl
TSNSCTL	
asysctl.h	ASysCtl_enableTemperatureSensor
asysctl.h	ASysCtl_disableTemperatureSensor
ANAREFCTL	
asysctl.h	ASysCtl_setVREF
VREGCTL	
asysctl.h	ASysCtl_enableVMONMask
asysctl.h	ASysCtl_powerDownVREG
VMONCTL	
asysctl.h	ASysCtl_enableBORL
asysctl.h	ASysCtl_disableBORL
CMPHPMXSEL	
asysctl.h	ASysCtl_configCMPMux
CMPHPMXSEL1	
-	
CMPLPMXSEL	
-	
CMPLPMXSEL1	
-	
CMPHNMXSEL	
-	
CMPLNMXSEL	
-	
LOCK	
asysctl.h	ASysCtl_lockTemperatureSensor
asysctl.h	ASysCtl_lockANAREF
asysctl.h	ASysCtl_lockVREG
asysctl.h	ASysCtl_lockVMON
asysctl.h	ASysCtl_lockCMPHPMux
asysctl.h	ASysCtl_lockCMPLPMux

Table 23-6. ASYSCTL Registers to Driverlib Functions (continued)

File	Driverlib Function
asysctl.h	ASysCtl_lockCMPHNMux
asysctl.h	ASysCtl_lockCMPLNMux
IODRVSEL	
asysctl.h	ASysCtl_setIOBufferDrive
IOMODESEL	
asysctl.h	ASysCtl_setIOBufferMode
AGPIOFILTER	
asysctl.h	ASysCtl_setAGPIOFilter
AGPIOCTRLH	
gpio.c	GPIO_setAnalogMode
PARITY_TEST	
-	
ADCSOCFRCGB	
asysctl.h	AsysCtl_configADCGlobalSOC
asysctl.h	AsysCtl_forceADCGlobalSOC
ADCSOCFRCGBSEL	
asysctl.h	AsysCtl_configADCGlobalSOC
PARITY_TEST_ALT1	
-	

23.7 Lock Registers

Setting one of the bits in the lock registers prevents further writes to the associated analog subsystem configuration register. This lock can only be cleared by a device reset.

23.8 ASBSYS Registers

This Section describes the ASBSYS Registers.

23.8.1 ASBSYS Base Address Table

Table 23-7. ASBSYS Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
ANALOG_SUBSYS_REGS [1]	ANALOGSUBSYS_BASE	0x3018_2000	-	YES	YES	YES	-	-	YES	YES

(1) Registers writeable by CPU1.LINK0, CPU1.LINK1, CPU1.LINK2 only. All CPUs can read all registers in all LINKs. Debug write access only allowed if Zone0 or Zone1 are enabled for full debug by all CPUs. Debug reads always allowed. Register Read/Write access by HSM.

23.8.2 ANALOG_SUBSYS_REGS Registers

Table 23-8 lists the memory-mapped registers for the ANALOG_SUBSYS_REGS registers. All register offset addresses not listed in Table 23-8 should be considered as reserved locations and the register contents should not be modified.

Table 23-8. ANALOG_SUBSYS_REGS Registers

Offset	Acronym	Register Name	Protection
44h	PMMVREGTRIM	Power Management Module VREG Trim Register	PARITY
70h	CTLTRIMSTS	HWCTL TRIM Error Status register	
154h	REFBUFCONFIGCDE	Config register for analog reference CDE	PARITY
1A8h	INTERNALTESTCTL	INTERNALTEST Node Control Register	PARITY
1E8h	CONFIGLOCK	Lock Register for all the config registers.	PARITY
1ECh	TSNSCTL	Temperature Sensor Control Register	PARITY
20Ch	ANAREFCTL	Analog Reference Control Register.	PARITY
214h	VREGCTL	Voltage Regulator Control Register	PARITY
22Ch	VMONCTL	Voltage Monitor Control Register	PARITY
26Ch	CMPHPMXSEL	Bits to select one of the many sources on CopmHP inputs. Refer to Pimux diagram for details.	PARITY
270h	CMPHPMXSEL1	Bits to select one of the many sources on CopmHP inputs. Refer to Pimux diagram for details.	PARITY
274h	CMPLPMXSEL	Bits to select one of the many sources on CopmLP inputs. Refer to Pimux diagram for details.	PARITY
278h	CMPLPMXSEL1	Bits to select one of the many sources on CopmLP inputs. Refer to Pimux diagram for details.	PARITY
27Ch	CMPHNMXSEL	Bits to select one of the many sources on CopmHN inputs. Refer to Pimux diagram for details.	PARITY
280h	CMPLNMXSEL	Bits to select one of the many sources on CopmLN inputs. Refer to Pimux diagram for details.	PARITY
294h	LOCK	Lock Register	PARITY
458h	IODRVSEL	5V FS IO Drive strength select register	PARITY
45Ch	IOMODESEL	PMBUS IO Mode select register	PARITY
464h	AGPIOFILTER	AGPIO Filter Control Register	PARITY
48Ch	AGPIOCTRLH	AGPIO Control Register	PARITY
4B4h	PARITY_TEST	Enables parity test	

Complex bit access types are encoded to fit into small table cells. Table 23-9 shows the codes that are used for access types in this section.

Table 23-9. ANALOG_SUBSYS_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WOnce	W Once	Write Write once

Table 23-9. ANALOG_SUBSYS_REGS Access Type Codes (continued)

Access Type	Code	Description
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

23.8.2.1 PMMVREGTRIM Register (Offset = 44h) [Reset = 00000000h]

PMMVREGTRIM is shown in Figure 23-4 and described in Table 23-10.

Return to the Summary Table.

Power Management Module VREG Trim Register

Figure 23-4. PMMVREGTRIM Register

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

Table 23-10. PMMVREGTRIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-0	VREGTRIM	R/W	0h	Core VDD Voltage Regulator Trim. This bit field defines the trim value for the core voltage regulator. = 0.9+ <VREGTRIM>*3.125mV Reset type: PORESETn

23.8.2.2 CTLTRIMSTS Register (Offset = 70h) [Reset = 00000000h]

CTLTRIMSTS is shown in Figure 23-5 and described in Table 23-11.

Return to the Summary Table.

HWCTL TRIM Error Status register

Figure 23-5. CTLTRIMSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED				RESERVED	INVKEY3	INVKEY2	INVKEY1
R-0-0h				R-0-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED				TIMEOUT	TWOBERR	RESERVED	SCANERR
R-0-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
SBERR	RESERVED						CTLSTS
R-0h	R-0-0h						R-0h

Table 23-11. CTLTRIMSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	Reserved
23-20	RESERVED	R-0	0h	Reserved
19	RESERVED	R-0	0h	Reserved
18	INVKEY3	R	0h	CTL_STATS : Invalid key read during TLC3 0x0 : No Invalid key read during TLC3 0x1 : Invalid key read during TLC3 detected Reset type: PORESETn
17	INVKEY2	R	0h	CTL_STATS : Invalid key read during TLC2 0x0 : No Invalid key read during TLC2 0x1 : Invalid key read during TLC2 detected Reset type: PORESETn
16	INVKEY1	R	0h	CTL_STATS : Invalid key read during TLC1 0x0 : No Invalid key read during TLC1 0x1 : Invalid key read during TLC1 detected Reset type: PORESETn
15-12	RESERVED	R-0	0h	Reserved
11	TIMEOUT	R	0h	CTL_STATS : TIMEOUT Error Detected 0x0 : No TIMEOUT Error detected 0x1 : TIMEOUT Error detected Once Reset gets released, trim load is expected to complete within a configured time (determined by a MAX_TRIM_LOAD_TIME tie-off). If the combined delay with the trim load and the various other delays (determined by tie-off values) is not complete within the MAX_TRIM_LOAD_TIME tie-off, timeout error is issued and trim load process exists Reset type: PORESETn
10	TWOBERR	R	0h	CTL_STATUS : Two bit error while reading from flash 0x0 : Two bit error not detected 0x1 : Two bit error detected Reset type: PORESETn
9	RESERVED	R	0h	Reserved

Table 23-11. CTLTRIMSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	SCANERR	R	0h	CTL_STATS : Wrong Scan Chain Error Detected 0x0 : No Error detected on Scan Chain 0x1 : Error detected on Scan Chain TLCx starts the scan chain with a signature 0xAA55CC33 and it expects to see the same after the scan chain is full. The autoload machine was able to fill the scan chain, but the wrong signature was returned. Reset type: PORESETn
7	SBERR	R	0h	CTL_STATUS : Single bit error while reading from flash 0x0 : Single bit error not detected 0x1 : Single bit error detected Reset type: PORESETn
6-1	RESERVED	R-0	0h	Reserved
0	CTLSTS	R	0h	CTL_STATUS : Trim Load complete 0x0 : TRIM Load not complete 0x1 : TRIM Load completed Reset type: PORESETn

23.8.2.3 REFBUFCONFIGCDE Register (Offset = 154h) [Reset = 00000000h]

REFBUFCONFIGCDE is shown in Figure 23-6 and described in Table 23-12.

Return to the Summary Table.

Config register for analog reference CDE

Figure 23-6. REFBUFCONFIGCDE Register

31	30	29	28	27	26	25	24
Spare1							
R/W-0h							
23	22	21	20	19	18	17	16
Spare1			DIS_ADCE_SP_SLEWBOOST	ADCE_TM_ENZ_DUTY_CYCLE	DIS_ADCD_SP_SLEWBOOST	ADCDTM_ENZ_DUTY_CYCLE	DIS_ADCC_SP_SLEWBOOST
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
ADCC_TM_ENZ_DUTY_CYCLE	ADCE_CHSEL_SOC_DEL_PROG	ADCE_MSB_RES_DAMP		ADCE_LSB_RES_DAMP		ADCD_CHSEL_SOC_DEL_PROG	ADCD_MSB_RES_DAMP
R/W-0h	R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
ADCD_MSB_RES_DAMP	ADCD_LSB_RES_DAMP		ADCC_CHSEL_SOC_DEL_PROG	ADCC_MSB_RES_DAMP		ADCC_LSB_RES_DAMP	
R/W-0h	R/W-0h		R/W-0h	R/W-0h		R/W-0h	

Table 23-12. REFBUFCONFIGCDE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	Spare1	R/W	0h	Connect to CIO Reset type: XRSn
20	DIS_ADCE_SP_SLEWBOOST	R/W	0h	0=Default SP slew boost scheme enabled 1=SP slew boost scheme disabled Reset type: XRSn
19	ADCE_TM_ENZ_DUTY_CYCLE	R/W	0h	Disables effect of PULSE_EXTENSION testmode Reset type: XRSn
18	DIS_ADCD_SP_SLEWBOOST	R/W	0h	0=Default SP slew boost scheme enabled 1=SP slew boost scheme disabled Reset type: XRSn
17	ADCDTM_ENZ_DUTY_CYCLE	R/W	0h	Disables effect of PULSE_EXTENSION testmode Reset type: XRSn
16	DIS_ADCC_SP_SLEWBOOST	R/W	0h	0=Default SP slew boost scheme enabled 1=SP slew boost scheme disabled Reset type: XRSn
15	ADCC_TM_ENZ_DUTY_CYCLE	R/W	0h	Disables effect of PULSE_EXTENSION testmode Reset type: XRSn
14	ADCE_CHSEL_SOC_DEL_PROG	R/W	0h	CHSEL rise to ADCSOC rise delay program Reset type: XRSn
13-12	ADCE_MSB_RES_DAMP	R/W	0h	ADC MSB damp res prog Reset type: XRSn
11-10	ADCE_LSB_RES_DAMP	R/W	0h	ADC LSB damp res prog Reset type: XRSn

Table 23-12. REFBUFCONFIGCDE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	ADCD_CHSEL_SOC_DEL_PROG	R/W	0h	CHSEL rise to ADCSOC rise delay program Reset type: XRSn
8-7	ADCD_MSB_RES_DAMP	R/W	0h	ADC MSB damp res prog Reset type: XRSn
6-5	ADCD_LSB_RES_DAMP	R/W	0h	ADC LSB damp res prog Reset type: XRSn
4	ADCC_CHSEL_SOC_DEL_PROG	R/W	0h	CHSEL rise to ADCSOC rise delay program Reset type: XRSn
3-2	ADCC_MSB_RES_DAMP	R/W	0h	ADC MSB damp res prog Reset type: XRSn
1-0	ADCC_LSB_RES_DAMP	R/W	0h	ADC LSB damp res prog Reset type: XRSn

23.8.2.4 INTERNALTESTCTL Register (Offset = 1A8h) [Reset = 00000000h]

INTERNALTESTCTL is shown in Figure 23-7 and described in Table 23-13.

Return to the Summary Table.

INTERNALTEST Node Control Register

Figure 23-7. INTERNALTESTCTL Register

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

Table 23-13. INTERNALTESTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0/W	0h	Reserved
15-12	RESERVED	R-0	0h	Reserved
11-9	RESERVED	R/W	0h	Reserved
8-6	RESERVED	R-0	0h	Reserved

Table 23-13. INTERNALTESTCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-0	TESTSEL	R/W	0h	Test Select. This bit field defines which internal node, if any, is selected to come out on the INTERNALTEST node connected to the ADC. Any values not defined below are reserved. Reset type: SYSRSn 0h (R/W) = No internal connection 1h (R/W) = Core VDD (1.2V) voltage 2h (R/W) = VDDA voltage 3h (R/W) = VSSA - Analog ground pin 4h (R/W) = VREFLOA pin voltage 5h (R/W) = CMPSS1 High DAC output (6-bit) 6h (R/W) = CMPSS1 Low DAC output (6-bit) 7h (R/W) = CMPSS2 High DAC output (6-bit) 8h (R/W) = CMPSS2 Low DAC output (6-bit) 9h (R/W) = CMPSS3 High DAC output (6-bit) Ah (R/W) = CMPSS3 Low DAC output (6-bit) Bh (R/W) = CMPSS4 High DAC output (6-bit) Ch (R/W) = CMPSS4 Low DAC output (6-bit) Dh (R/W) = CMPSS5 High DAC output (6-bit) Eh (R/W) = CMPSS5 Low DAC output (6-bit) Fh (R/W) = CMPSS6 High DAC output (6-bit) 10h (R/W) = CMPSS6 Low DAC output (6-bit) 11h (R/W) = CMPSS7 High DAC output (6-bit) 12h (R/W) = CMPSS7 Low DAC output (6-bit) 13h (R/W) = CMPSS8 High DAC output (6-bit) 14h (R/W) = CMPSS8 Low DAC output (6-bit) 15h (R/W) = CMPSS9 High DAC output (6-bit) 16h (R/W) = CMPSS9 Low DAC output (6-bit) 17h (R/W) = CMPSS10 High DAC output (6-bit) 18h (R/W) = CMPSS10 Low DAC output (6-bit) 19h (R/W) = CMPSS11 High DAC output (6-bit) 1Ah (R/W) = CMPSS11 Low DAC output (6-bit) 1Bh (R/W) = CMPSS11 High DAC output (6-bit) 1Ch (R/W) = CMPSS11 Low DAC output (6-bit) 1Dh (R/W) = Enable ENZ_CALIB_GAIN_3P3V. All ADCs are placed in gain calibration mode. 0.9*VREFHIAB pin voltage is sampled by all ADCs through INTERNALTEST mux output, overriding CHSEL setting. 1Eh (R/W) = Reserved 1Fh (R/W) = Reserved 20h (R/W) = Reserved 21h (R/W) = Reserved 22h (R/W) = Reserved 23h (R/W) = Reserved 24h (R/W) = Reserved 25h (R/W) = Reserved 26h (R/W) = Reserved 27h (R/W) = Reserved 28h (R/W) = Reserved 29h (R/W) = Reserved 2Ah (R/W) = Reserved 2Bh (R/W) = Reserved 2Ch (R/W) = VSS - Digital ground pin 2Dh (R/W) = Reserved 2Eh (R/W) = Reserved 2Fh (R/W) = Reserved 30h (R/W) = Reserved 31h (R/W) = VREFLOCDE pin voltage

23.8.2.5 CONFIGLOCK Register (Offset = 1E8h) [Reset = 00000000h]

CONFIGLOCK is shown in Figure 23-8 and described in Table 23-14.

Return to the Summary Table.

Lock Register for all the config registers.

Figure 23-8. CONFIGLOCK Register

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

Table 23-14. CONFIGLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R-0	0h	Reserved
6	RESERVED	R/WSonce	0h	Reserved
5	RESERVED	R/WSonce	0h	Reserved
4	RESERVED	R/WSonce	0h	Reserved
3	AGPIOCTRL	R/WSonce	0h	Locks all AGPIOCTRL Register. Setting this bit will disable any future writes to this register. This bit can only be cleared by a reset. Reset type: SYSRSn
2	RESERVED	R/WSonce	0h	Reserved
1	AGPIOFILTER	R/WSonce	0h	Locks AGPIOFILTER Register. Setting this bit will disable any future writes to this register. This bit can only be cleared by a reset. Reset type: SYSRSn
0	LOCKBIT	R/WSonce	0h	Locks INTERNALTESTCTL register and other analog subsystem configuration settings. Setting this bit will disable any future writes to this register. This bit can only be cleared by a reset. Reset type: SYSRSn

23.8.2.6 TSNSCTL Register (Offset = 1ECh) [Reset = 00000000h]

TSNSCTL is shown in Figure 23-9 and described in Table 23-15.

Return to the Summary Table.

Temperature Sensor Control Register

Figure 23-9. TSNSCTL Register

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

Table 23-15. TSNSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-1	RESERVED	R-0	0h	Reserved
0	ENABLE	R/W	0h	Temperature Sensor Enable. This bit enables the temperature sensor output to the ADC. 0 Disabled 1 Enabled Reset type: SYSRSn

23.8.2.7 ANAREFCTL Register (Offset = 20Ch) [Reset = 0000003Fh]

ANAREFCTL is shown in Figure 23-10 and described in Table 23-16.

Return to the Summary Table.

Analog Reference Control Register.

Figure 23-10. ANAREFCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED			RESERVED	ANAREFCDE_2P5SEL	ANAREFAB_2P5SEL
R/W-0h	R/W-0h	R-0-0h			R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED			RESERVED	RESERVED	RESERVED	ANAREFCDESEL	ANAREFABSEL
R-0-1h			R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 23-16. ANAREFCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13-11	RESERVED	R-0	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	ANAREFCDE_2P5SEL	R/W	0h	Analog reference B 2.5V to Analog reference CDE 2.5V. In internal reference mode, this bit selects which voltage the internal reference buffer drives onto the VREFHI pin. The buffer can drive either 1.65V onto the pin, resulting in a reference range of 0 to 3.3V, or the buffer can drive 2.5V onto the pin, resulting in a reference range of 0 to 2.5V. If switching between these two modes, the user must allow adequate time for the external capacitor to charge to the new voltage before using the ADC or buffered DAC. If multiple VREFHI pins are ganged together (for lower pin-count packages), then the reference voltage select for the ganged pins should always be configured to the same setting. 0 Internal 1.65V reference mode (3.3V reference range) 1 Internal 2.5V reference mode (2.5V reference range) Reset type: XRSn

Table 23-16. ANAREFCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	ANAREFAB_2P5SEL	R/W	0h	Analog reference A 2.5V to Analog reference AB 2.5V. In internal reference mode, this bit selects which voltage the internal reference buffer drives onto the VREFHI pin. The buffer can drive either 1.65V onto the pin, resulting in a reference range of 0 to 3.3V, or the buffer can drive 2.5V onto the pin, resulting in a reference range of 0 to 2.5V. If switching between these two modes, the user must allow adequate time for the external capacitor to charge to the new voltage before using the ADC or buffered DAC. If multiple VREFHI pins are ganged together (for lower pin-count packages), then the reference voltage select for the ganged pins should always be configured to the same setting. 0 Internal 1.65V reference mode (3.3V reference range) 1 Internal 2.5V reference mode (2.5V reference range) Reset type: XRSn
7-5	RESERVED	R-0	1h	Reserved
4	RESERVED	R/W	1h	Reserved
3	RESERVED	R/W	1h	Reserved
2	RESERVED	R/W	1h	Reserved
1	ANAREFCDESEL	R/W	1h	Analog reference C/D/E mode select. This bit selects whether the VREFHICDE pin uses internal reference mode (the device drives a voltage onto the VREFHI pin) or external reference mode (the system is expected to drive a voltage into the VREFHI pin). If multiple VREFHI pins are ganged together (for lower pin-count packages), then the mode select for the ganged pins should always be configured to the same setting 0 Internal reference mode 1 External reference mode Reset type: XRSn
0	ANAREFABSEL	R/W	1h	Analog reference A/B mode select. This bit selects whether the VREFHIAB pin uses internal reference mode (the device drives a voltage onto the VREFHI pin) or external reference mode (the system is expected to drive a voltage into the VREFHI pin). If multiple VREFHI pins are ganged together (for lower pin-count packages), then the mode select for the ganged pins should always be configured to the same setting 0 Internal reference mode 1 External reference mode Reset type: XRSn

23.8.2.8 VREGCTL Register (Offset = 214h) [Reset = 00000000h]

VREGCTL is shown in Figure 23-11 and described in Table 23-17.

Return to the Summary Table.

Voltage Regulator Control Register

Figure 23-11. VREGCTL Register

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

Table 23-17. VREGCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	ENMASK	R/W	0h	Enable VMON function mask on a TRIM load. This bit, when set, masks the generation of POR/BOR for a fixed duration whenever TRIM bits are changed. If this bit is cleared, then POR/BOR generation is not masked on a TRIM load. Reset type: XRSn
14-1	RESERVED	R-0	0h	Reserved
0	PWRDNVREG	R/W	0h	Power down internal voltage regulator. This bit, when cleared, enables the internal voltage regulator, when set powers down the internal voltage regulator. Note: On devices, in which VREGENZ is not bonded out, this bit should be used to disable internal VREG. Reset type: XRSn

23.8.2.9 VMONCTL Register (Offset = 22Ch) [Reset = 00000000h]

VMONCTL is shown in Figure 23-12 and described in Table 23-18.

Return to the Summary Table.

Voltage Monitor Control Register

Figure 23-12. VMONCTL Register

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

Table 23-18. VMONCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-9	RESERVED	R-0	0h	Reserved
8	BORLVMONDIS	R/W	0h	BORL disable on VDDIO. 0 BORL is enabled on VDDIO, i.e BOR circuit will be triggered if VDDIO goes lower than the lower BOR threshold of VDDIO. 1 BORL is disabled on VDDIO, i.e BOR circuit will not be triggered if VDDIO goes lower than the lower BOR threshold of VDDIO. Reset type: SYSRSn
7-2	RESERVED	R-0	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

23.8.2.10 CMPHPMXSEL Register (Offset = 26Ch) [Reset = 00000000h]

CMPHPMXSEL is shown in Figure 23-13 and described in Table 23-19.

Return to the Summary Table.

Bits to select one of the many sources on CopmHP inputs. Refer to Pimux diagram for details.

Figure 23-13. CMPHPMXSEL Register

31	30	29	28	27	26	25	24
RESERVED		CMP10HPMXSEL			CMP9HPMXSEL		
R-0-0h		R/W-0h			R/W-0h		
23	22	21	20	19	18	17	16
CMP8HPMXSEL			CMP7HPMXSEL			CMP6HPMXSEL	
R/W-0h			R/W-0h			R/W-0h	
15	14	13	12	11	10	9	8
CMP6HPMXSEL	CMP5HPMXSEL			CMP4HPMXSEL			CMP3HPMXSEL
R/W-0h	R/W-0h			R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
CMP3HPMXSEL		CMP2HPMXSEL			CMP1HPMXSEL		
R/W-0h		R/W-0h			R/W-0h		

Table 23-19. CMPHPMXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R-0	0h	Reserved
29-27	CMP10HPMXSEL	R/W	0h	CMP10HPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
26-24	CMP9HPMXSEL	R/W	0h	CMP9HPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
23-21	CMP8HPMXSEL	R/W	0h	CMP8HPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
20-18	CMP7HPMXSEL	R/W	0h	CMP7HPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
17-15	CMP6HPMXSEL	R/W	0h	CMP6HPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
14-12	CMP5HPMXSEL	R/W	0h	CMP5HPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
11-9	CMP4HPMXSEL	R/W	0h	CMP4HPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn

Table 23-19. CMPHPMXSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8-6	CMP3HPMXSEL	R/W	0h	CMP3HPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
5-3	CMP2HPMXSEL	R/W	0h	CMP2HPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
2-0	CMP1HPMXSEL	R/W	0h	CMP1HPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn

23.8.2.11 CMPHPMXSEL1 Register (Offset = 270h) [Reset = 00000000h]

CMPHPMXSEL1 is shown in Figure 23-14 and described in Table 23-20.

Return to the Summary Table.

Bits to select one of the many sources on CopmHP inputs. Refer to Pimux diagram for details.

Figure 23-14. CMPHPMXSEL1 Register

31	30	29	28	27	26	25	24
RESERVED		RESERVED			RESERVED		
R-0-0h		R/W-0h			R/W-0h		
23	22	21	20	19	18	17	16
RESERVED			RESERVED			RESERVED	
R/W-0h			R/W-0h			R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	RESERVED			RESERVED			RESERVED
R/W-0h	R/W-0h			R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
RESERVED		CMP12HPMXSEL			CMP11HPMXSEL		
R/W-0h		R/W-0h			R/W-0h		

Table 23-20. CMPHPMXSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R-0	0h	Reserved
29-27	RESERVED	R/W	0h	Reserved
26-24	RESERVED	R/W	0h	Reserved
23-21	RESERVED	R/W	0h	Reserved
20-18	RESERVED	R/W	0h	Reserved
17-15	RESERVED	R/W	0h	Reserved
14-12	RESERVED	R/W	0h	Reserved
11-9	RESERVED	R/W	0h	Reserved
8-6	RESERVED	R/W	0h	Reserved
5-3	CMP12HPMXSEL	R/W	0h	CMP12HPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
2-0	CMP11HPMXSEL	R/W	0h	CMP11HPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn

23.8.2.12 CMPLPMXSEL Register (Offset = 274h) [Reset = 00000000h]

CMPLPMXSEL is shown in Figure 23-15 and described in Table 23-21.

Return to the Summary Table.

Bits to select one of the many sources on CopmLP inputs. Refer to Pimux diagram for details.

Figure 23-15. CMPLPMXSEL Register

31	30	29	28	27	26	25	24
RESERVED		CMP10LPMXSEL			CMP9LPMXSEL		
R-0-0h		R/W-0h			R/W-0h		
23	22	21	20	19	18	17	16
CMP8LPMXSEL			CMP7LPMXSEL			CMP6LPMXSEL	
R/W-0h			R/W-0h			R/W-0h	
15	14	13	12	11	10	9	8
CMP6LPMXSEL	CMP5LPMXSEL			CMP4LPMXSEL			CMP3LPMXSEL
R/W-0h	R/W-0h			R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
CMP3LPMXSEL		CMP2LPMXSEL			CMP1LPMXSEL		
R/W-0h		R/W-0h			R/W-0h		

Table 23-21. CMPLPMXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R-0	0h	Reserved
29-27	CMP10LPMXSEL	R/W	0h	CMP10LPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
26-24	CMP9LPMXSEL	R/W	0h	CMP9LPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
23-21	CMP8LPMXSEL	R/W	0h	CMP8LPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
20-18	CMP7LPMXSEL	R/W	0h	CMP7LPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
17-15	CMP6LPMXSEL	R/W	0h	CMP6LPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
14-12	CMP5LPMXSEL	R/W	0h	CMP5LPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
11-9	CMP4LPMXSEL	R/W	0h	CMP4LPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn

Table 23-21. CMPLPMXSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8-6	CMP3LPMXSEL	R/W	0h	CMP3LPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
5-3	CMP2LPMXSEL	R/W	0h	CMP2LPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
2-0	CMP1LPMXSEL	R/W	0h	CMP1LPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn

23.8.2.13 CMPLPMXSEL1 Register (Offset = 278h) [Reset = 00000000h]

CMPLPMXSEL1 is shown in Figure 23-16 and described in Table 23-22.

Return to the Summary Table.

Bits to select one of the many sources on CopmLP inputs. Refer to Pimux diagram for details.

Figure 23-16. CMPLPMXSEL1 Register

31	30	29	28	27	26	25	24
RESERVED		RESERVED			RESERVED		
R-0-0h		R/W-0h			R/W-0h		
23	22	21	20	19	18	17	16
RESERVED			RESERVED			RESERVED	
R/W-0h			R/W-0h			R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	RESERVED			RESERVED			RESERVED
R/W-0h	R/W-0h			R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
RESERVED		CMP12LPMXSEL			CMP11LPMXSEL		
R/W-0h		R/W-0h			R/W-0h		

Table 23-22. CMPLPMXSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R-0	0h	Reserved
29-27	RESERVED	R/W	0h	Reserved
26-24	RESERVED	R/W	0h	Reserved
23-21	RESERVED	R/W	0h	Reserved
20-18	RESERVED	R/W	0h	Reserved
17-15	RESERVED	R/W	0h	Reserved
14-12	RESERVED	R/W	0h	Reserved
11-9	RESERVED	R/W	0h	Reserved
8-6	RESERVED	R/W	0h	Reserved
5-3	CMP12LPMXSEL	R/W	0h	CMP12LPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn
2-0	CMP11LPMXSEL	R/W	0h	CMP11LPMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Note: Only values 0 to 4 are valid, rest are reserved Reset type: XRSn

23.8.2.14 CMPHNMXSEL Register (Offset = 27Ch) [Reset = 00000000h]

CMPHNMXSEL is shown in Figure 23-17 and described in Table 23-23.

Return to the Summary Table.

Bits to select one of the many sources on CopmHN inputs. Refer to Pimux diagram for details.

Figure 23-17. CMPHNMXSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				CMP12HNMXSEL	CMP11HNMXSEL	CMP10HNMXSEL	CMP9HNMXSEL
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CMP8HNMXSEL	CMP7HNMXSEL	CMP6HNMXSEL	CMP5HNMXSEL	CMP4HNMXSEL	CMP3HNMXSEL	CMP2HNMXSEL	CMP1HNMXSEL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 23-23. CMPHNMXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-12	RESERVED	R-0	0h	Reserved
11	CMP12HNMXSEL	R/W	0h	CMP12HNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
10	CMP11HNMXSEL	R/W	0h	CMP11HNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
9	CMP10HNMXSEL	R/W	0h	CMP10HNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
8	CMP9HNMXSEL	R/W	0h	CMP9HNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
7	CMP8HNMXSEL	R/W	0h	CMP8HNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
6	CMP7HNMXSEL	R/W	0h	CMP7HNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
5	CMP6HNMXSEL	R/W	0h	CMP6HNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
4	CMP5HNMXSEL	R/W	0h	CMP5HNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn

Table 23-23. CMPHNMXSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	CMP4HNMXSEL	R/W	0h	CMP4HNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
2	CMP3HNMXSEL	R/W	0h	CMP3HNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
1	CMP2HNMXSEL	R/W	0h	CMP2HNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
0	CMP1HNMXSEL	R/W	0h	CMP1HNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn

23.8.2.15 CMPLNMXSEL Register (Offset = 280h) [Reset = 00000000h]

CMPLNMXSEL is shown in Figure 23-18 and described in Table 23-24.

Return to the Summary Table.

Bits to select one of the many sources on CopmLN inputs. Refer to Pimux diagram for details.

Figure 23-18. CMPLNMXSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				CMP12LNMXSEL	CMP11LNMXSEL	CMP10LNMXSEL	CMP9LNMXSEL
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CMP8LNMXSEL	CMP7LNMXSEL	CMP6LNMXSEL	CMP5LNMXSEL	CMP4LNMXSEL	CMP3LNMXSEL	CMP2LNMXSEL	CMP1LNMXSEL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 23-24. CMPLNMXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-12	RESERVED	R-0	0h	Reserved
11	CMP12LNMXSEL	R/W	0h	CMP12LNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
10	CMP11LNMXSEL	R/W	0h	CMP11LNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
9	CMP10LNMXSEL	R/W	0h	CMP10LNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
8	CMP9LNMXSEL	R/W	0h	CMP9LNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
7	CMP8LNMXSEL	R/W	0h	CMP8LNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
6	CMP7LNMXSEL	R/W	0h	CMP7LNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
5	CMP6LNMXSEL	R/W	0h	CMP6LNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
4	CMP5LNMXSEL	R/W	0h	CMP5LNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn

Table 23-24. CMPLNMXSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	CMP4LNMXSEL	R/W	0h	CMP4LNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
2	CMP3LNMXSEL	R/W	0h	CMP3LNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
1	CMP2LNMXSEL	R/W	0h	CMP2LNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn
0	CMP1LNMXSEL	R/W	0h	CMP1LNMXSEL bits, Refer to analog grouping connection in the Analog Chapter of TRM Reset type: XRSn

23.8.2.16 LOCK Register (Offset = 294h) [Reset = 00000000h]

LOCK is shown in Figure 23-19 and described in Table 23-25.

Return to the Summary Table.

Lock Register

Figure 23-19. LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED			CMPLPMXSEL1	CMPHPMXSEL1	RESERVED	VREGCTL	CMPLNMXSEL
R-0-0h			R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
CMPHNMXSEL	CMPLPMXSEL	CMPHPMXSEL	RESERVED	RESERVED	VMONCTL	ANAREFCTL	TSNSCTL
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 23-25. LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R-0	0h	Reserved
12	CMPLPMXSEL1	R/WSonce	0h	CMPLPMXSEL1 Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn
11	CMPHPMXSEL1	R/WSonce	0h	CMPHPMXSEL1 Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn
10	RESERVED	R/WSonce	0h	Reserved
9	VREGCTL	R/WSonce	0h	VREGCTL Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn
8	CMPLNMXSEL	R/WSonce	0h	CMPLNMXSEL Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn
7	CMPHNMXSEL	R/WSonce	0h	CMPHNMXSEL Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn
6	CMPLPMXSEL	R/WSonce	0h	CMPLPMXSEL Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn
5	CMPHPMXSEL	R/WSonce	0h	CMPHPMXSEL Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn

Table 23-25. LOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	RESERVED	R/WSonce	0h	Reserved
3	RESERVED	R/WSonce	0h	Reserved
2	VMONCTL	R/WSonce	0h	VMONCTL Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn
1	ANAREFCTL	R/WSonce	0h	ANAREFCTL Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn
0	TSNSCTL	R/WSonce	0h	TSNSCTL Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn

23.8.2.17 IODRVSEL Register (Offset = 458h) [Reset = 00000000h]

IODRVSEL is shown in Figure 23-20 and described in Table 23-26.

Return to the Summary Table.

5V FS IO Drive strength select register

Figure 23-20. IODRVSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		GPIO29	GPIO23	GPIO22	GPIO18	GPIO15	GPIO10
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 23-26. IODRVSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	GPIO29	R/W	0h	Drive Select for the IO buffer: 0: IO will have 4ma drive (default) 1: IO will support 20ma drive for PMBUS operation Reset type: XRSn
4	GPIO23	R/W	0h	Drive Select for the IO buffer: 0: IO will have 4ma drive (default) 1: IO will support 20ma drive for PMBUS operation Reset type: XRSn
3	GPIO22	R/W	0h	Drive Select for the IO buffer: 0: IO will have 4ma drive (default) 1: IO will support 20ma drive for PMBUS operation Reset type: XRSn
2	GPIO18	R/W	0h	Drive Select for the IO buffer: 0: IO will have 4ma drive (default) 1: IO will support 20ma drive for PMBUS operation Reset type: XRSn
1	GPIO15	R/W	0h	Drive Select for the IO buffer: 0: IO will have 4ma drive (default) 1: IO will support 20ma drive for PMBUS operation Reset type: XRSn
0	GPIO10	R/W	0h	Drive Select for the IO buffer: 0: IO will have 4ma drive (default) 1: IO will support 20ma drive for PMBUS operation Reset type: XRSn

23.8.2.18 IOMODESEL Register (Offset = 45Ch) [Reset = 00000000h]

IOMODESEL is shown in Figure 23-21 and described in Table 23-27.

Return to the Summary Table.

PMBUS IO Mode select register

Figure 23-21. IOMODESEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		GPIO29	GPIO23	GPIO22	GPIO18	GPIO15	GPIO10
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 23-27. IOMODESEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	GPIO29	R/W	0h	Mode Select for the IO buffer: 0: IO buffer is set to operate at 3.3v (default) 1: IO buffer is set to operate at 1.35v for PMBUS Reset type: XRSn
4	GPIO23	R/W	0h	Mode Select for the IO buffer: 0: IO buffer is set to operate at 3.3v (default) 1: IO buffer is set to operate at 1.35v for PMBUS Reset type: XRSn
3	GPIO22	R/W	0h	Mode Select for the IO buffer: 0: IO buffer is set to operate at 3.3v (default) 1: IO buffer is set to operate at 1.35v for PMBUS Reset type: XRSn
2	GPIO18	R/W	0h	Mode Select for the IO buffer: 0: IO buffer is set to operate at 3.3v (default) 1: IO buffer is set to operate at 1.35v for PMBUS Reset type: XRSn
1	GPIO15	R/W	0h	Mode Select for the IO buffer: 0: IO buffer is set to operate at 3.3v (default) 1: IO buffer is set to operate at 1.35v for PMBUS Reset type: XRSn
0	GPIO10	R/W	0h	Mode Select for the IO buffer: 0: IO buffer is set to operate at 3.3v (default) 1: IO buffer is set to operate at 1.35v for PMBUS Reset type: XRSn

23.8.2.19 AGPIOFILTER Register (Offset = 464h) [Reset = 00000000h]

AGPIOFILTER is shown in Figure 23-22 and described in Table 23-28.

Return to the Summary Table.

AGPIO Filter Control Register

Figure 23-22. AGPIOFILTER Register

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

Table 23-28. AGPIOFILTER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-10	RESERVED	R-0	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-2	RESERVED	R-0	0h	Reserved
1-0	FILTER	R/W	0h	AGPIOFILTER Control for filter side pins 00 : Filter bypass configuration (min ACQPS = 90ns) 01 : 333Ohm filter setting (min ACQPS = 125ns) 10 : 500 Ohm filter setting (min ACQPS = 160ns) 11 : 1KOhm filter setting (min ACQPS = 230ns) Reset type: XRSn

23.8.2.20 AGPIOCTRLH Register (Offset = 48Ch) [Reset = 00000000h]

AGPIOCTRLH is shown in Figure 23-23 and described in Table 23-29.

Return to the Summary Table.

AGPIO Control Register

Figure 23-23. AGPIOCTRLH Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GPIO249	GPIO248
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO247	GPIO246	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	GPIO240
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	GPIO236	GPIO235	GPIO234	GPIO233	GPIO232
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	GPIO229	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 23-29. AGPIOCTRLH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	GPIO249	R/W	0h	One time configuration for GPIO249 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
24	GPIO248	R/W	0h	One time configuration for GPIO248 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
23	GPIO247	R/W	0h	One time configuration for GPIO247 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
22	GPIO246	R/W	0h	One time configuration for GPIO246 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
21	GPIO245	R/W	0h	One time configuration for GPIO245 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn

Table 23-29. AGPIOCTRLH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO244	R/W	0h	One time configuration for GPIO244 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
19	GPIO243	R/W	0h	One time configuration for GPIO243 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
18	GPIO242	R/W	0h	One time configuration for GPIO242 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
17	GPIO241	R/W	0h	One time configuration for GPIO241 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
16	GPIO240	R/W	0h	One time configuration for GPIO240 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
15	GPIO239	R/W	0h	One time configuration for GPIO239 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
14	GPIO238	R/W	0h	One time configuration for GPIO238 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
13	GPIO237	R/W	0h	One time configuration for GPIO237 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
12	GPIO236	R/W	0h	One time configuration for GPIO236 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
11	GPIO235	R/W	0h	One time configuration for GPIO235 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
10	GPIO234	R/W	0h	One time configuration for GPIO234 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn

Table 23-29. AGPIOCTRLH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	GPIO233	R/W	0h	One time configuration for GPIO233 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
8	GPIO232	R/W	0h	One time configuration for GPIO232 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
7	GPIO231	R/W	0h	One time configuration for GPIO231 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
6	GPIO230	R/W	0h	One time configuration for GPIO230 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
5	GPIO229	R/W	0h	One time configuration for GPIO229 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
4	GPIO228	R/W	0h	One time configuration for GPIO228 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
3	GPIO227	R/W	0h	One time configuration for GPIO227 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
2	GPIO226	R/W	0h	One time configuration for GPIO226 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
1	GPIO225	R/W	0h	One time configuration for GPIO225 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
0	GPIO224	R/W	0h	One time configuration for GPIO224 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn

23.8.2.21 PARITY_TEST Register (Offset = 4B4h) [Reset = 00000000h]

PARITY_TEST is shown in Figure 23-24 and described in Table 23-30.

Return to the Summary Table.

Enables parity test

Figure 23-24. PARITY_TEST Register

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

Table 23-30. PARITY_TEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	TESTEN	R/W	0h	1010: Parity test feature is enabled All other values: Parity test feature is disabled Note: (1) When the parity test feature is enabled, actual registers are not accessible in the memory map. Instead, the parity values (final XOR output indicating the parity error) are accessible. Parity is computed for every byte and the corresponding parity error value is available at the bit-0 of every byte. Value of '1' written to the parity bit after enabling the parity test feature can be used to inject the error by inverting the stored parity value. (2) It is recommended to leave the field as 0101 or 0000 after completing the parity test. Reset type: SYSRSn

Chapter 24

Analog-to-Digital Converter (ADC)

The analog-to-digital converter (ADC) module described in this chapter is a . See the *C2000 Real-Time Control Peripheral Reference Guide* for a list of all devices with modules of the same type, to determine the differences between the types, and for a list of device-specific differences within a type.

24.1 Introduction

The ADC module is a 12-bit successive approximation (SAR) style ADC . The ADC module in this device has five instances of SAR ADC, two of which are designed with selectable resolution of either 16 bits or 12 bits, and three of which have a fixed resolution of 12 bits.The ADC is composed of a core and a wrapper. The core is composed of the analog circuits which include the channel select MUX, the sample-and-hold (S/H) circuit, the successive approximation circuits, voltage reference circuits, and other analog support circuits. The wrapper is composed of the digital circuits that configure and control the ADC. These circuits include the logic for programmable conversions, result registers, interfaces to analog circuits, interfaces to the peripheral buses, post-processing circuits, and interfaces to other on-chip modules.

Each ADC module consists of a single sample-and-hold (S/H) circuit. The ADC module is designed to be duplicated multiple times on the same chip, allowing simultaneous sampling or independent operation of multiple ADCs. The ADC wrapper is start-of-conversion (SOC) based (see Section 24.3).

24.1.1 ADC Related Collateral

Foundational Materials

- *ADC Input Circuit Evaluation for C2000 MCUs (PSPICE for TI) Application Report*
- *ADC Input Circuit Evaluation for C2000 MCUs (TINA-TI) Application Report*
- C28x Academy - ADC
- C29x Academy - Analog to Digital Converter (ADC)
- PSpice for TI design and simulation tool
- *Real-Time Control Reference Guide*
  - Refer to the ADC section
- TI Precision Labs - ADCs
- TI Precision Labs: Driving the reference input on a SAR ADC (Video)
- TI Precision Labs: Introduction to analog-to-digital converters (ADCs) (Video)
- TI Precision Labs: SAR ADC input driver design (Video)
- TI e2e: Connecting VDDA to VREFHI
- TI e2e: Topologies for ADC Input Protection
- TI e2e: Why does the ADC Input Voltage drop with sampling?
  - Sampling a high impedance voltage divider with ADC
- *Understanding Data Converters Application Report*
- *VREFHI Input Driver Design for C2000 MCUs Application Report*

Getting Started Materials

- ADC-PWM Synchronization Using ADC Interrupt
  - NOTE: This is a non-TI (third party) site.
- Analog-to-Digital Converter (ADC) Training for C2000 MCUs (Video)
- *Hardware Design Guide for F2800x C2000 Real-Time MCU Series*

Expert Materials

- *ADC Oversampling Application Report*
- Analog Engineer's Calculator
- *Analog Engineer's Pocket Reference*
- *Charge-Sharing Driving Circuits for C2000 ADCs (using PSPICE-FOR-TI) Application Report*
- *Charge-Sharing Driving Circuits for C2000 ADCs (using TINA-TI) Application Report*
- Debugging an integrated ADC in a microcontroller using an oscilloscope
- Hardware oversampling using C2000 ADC (Video)
- *Methods for Mitigating ADC Memory Cross-Talk Application Report*
- TI Precision Labs: ADC AC specifications (Video)
- TI Precision Labs: ADC Error sources (Video)
- TI Precision Labs: ADC Noise (Video)

- TI Precision Labs: Analog-to-digital converter (ADC) drive topologies (Video)
- TI Precision Labs: Electrical overstress on data converters (Video)
- TI Precision Labs: High-speed ADC fundamentals (Video)
- TI Precision Labs: SAR & Delta-Sigma: Understanding the Difference (Video)
- TI e2e: ADC Bandwidth Clarification
- TI e2e: ADC Resolution with Oversampling
- TI e2e: ADC configuration for interleaved mode
- TI e2e: Simultaneous Sampling with Single ADC

24.1.2 Features

Each ADC has the following features:

- Selectable resolution of 12 bits or 16 bits (ADC-A, ADC-B); 12-bit resolution (ADC-C, ADC-D, ADC-E)
- Ratiometric external reference set by VREFHI and VREFLO pins
- Selectable internal reference of 2.5V or 3.3V
- Differential signal conversions on ADC-A and ADC-B
- Single-ended signal conversions
- Input multiplexer with up to 80 channels (single-ended) or 40 channels (differential)
- External channel mux option to expand available ADC channels
- 32 configurable SOCs
- 32 individually addressable result registers
- Two trigger repeater modules, enabling customizable hardware oversampling and undersampling modes with little or no CPU overhead
- Multiple trigger sources
  - S/W (with available global synchronization for multiple ADCs) - software immediate start
  - All ePWMs- ADCSOC A or B
  - GPIO XINT2
  - CPU Timers 0/1/2 (from each C29 core present)
  - ADCINT1/2
  - ECAP events in capture mode (CEVT1, CEVT2, CEVT3, and CEVT4) and APWM mode (period match, compare match, or both)
- Four flexible PIPE interrupts
- Configurable interrupt placement
- Burst mode
- Four post-processing blocks, each with:
  - Saturating offset calibration
  - Error from set-point calculation
  - High, low, and zero-crossing compare, with interrupt and ePWM trip capability
  - Trigger-to-sample delay capture
  - Aggregation functions: max, min, sum, and average (binary shift)
  - Configurable digital filter for high/low/zero-crossing compare
  - Absolute value function
- Result safety checkers to compare SOC results on same ADC or multiple ADC instances

Note

Not every channel is pinned out from all ADCs. Check the device data sheet to determine which channels are available.

24.1.3 Block Diagram

Figure 24-1 shows the block diagram for the ADC core and ADC wrapper.

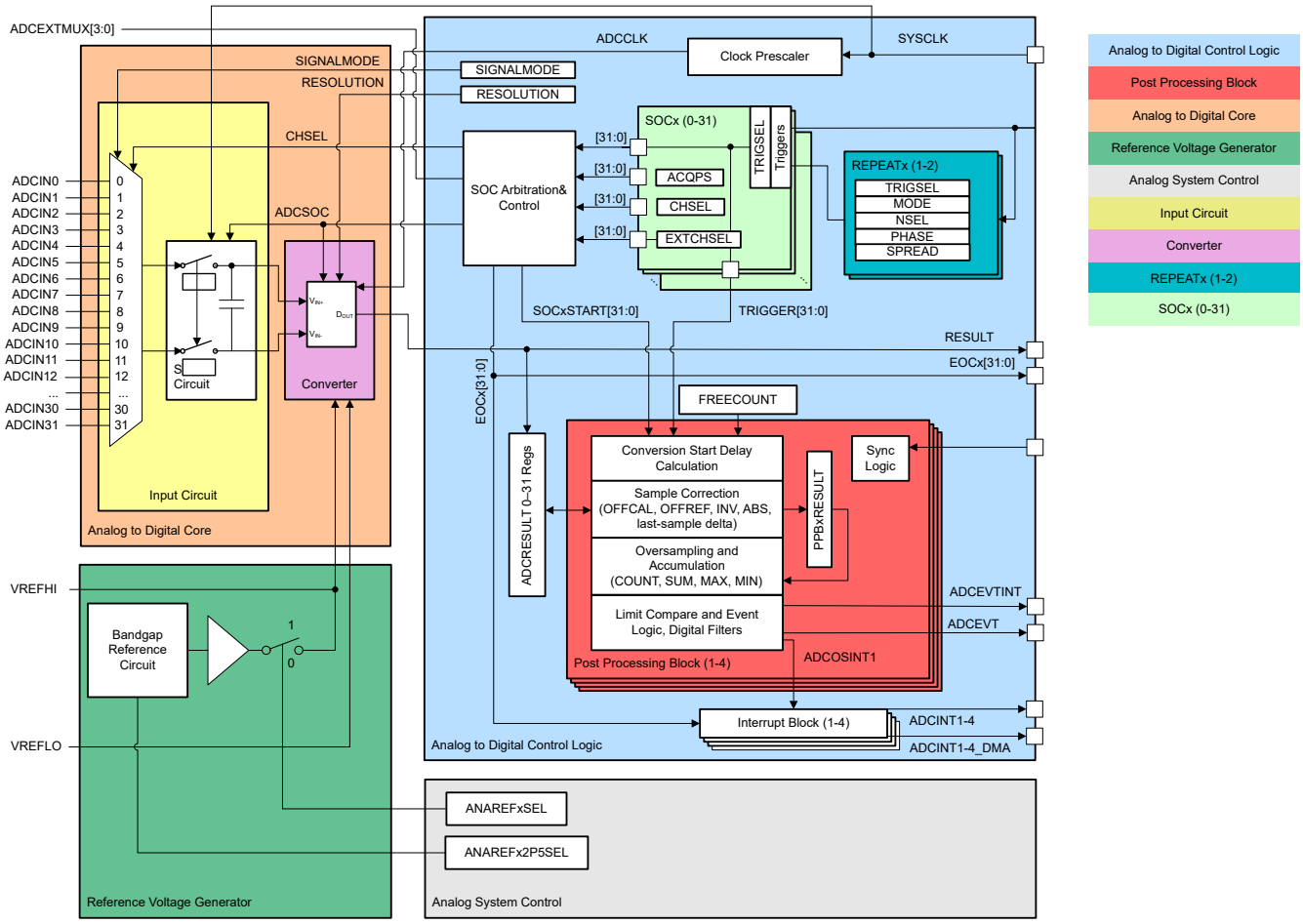


Figure 24-1. ADC Module Block Diagram

Note

- The ADC block diagram reflects the number of ADC channels internally configurable on the device. The actual number of available external ADC inputs varies depending on part and package.

24.2 ADC Configurability

Some ADC configurations are individually controlled by the SOCs, while others are globally controlled per ADC module. Table 24-1 summarizes the basic ADC options and the level of configurability. The subsequent sections discuss these configurations.

Table 24-1. ADC Options and Configuration Levels

Options	Configurability
Clock	Per module(1)
Resolution	Per module (ADC-A and ADC-B only) (1)
Signal mode	Per module (ADC-A and ADC-B only)
Reference voltage source	Per module (external or internal)(2)
Trigger source	Per SOC(1)
Converted channel	Per SOC
Acquisition window duration	Per SOC(1)
EOC location	Per module
Burst Mode	Per module(1)

(1) Writing these values differently to different ADC modules can cause the ADCs to operate asynchronously. See Section 24.14.1 for guidance on when the ADCs are operating synchronously or asynchronously.
(2) On this device, VREFHI pins are shared among multiple ADCs. In this case, the ADCs that share a reference pin must have the reference modes configured identically

24.2.1 Clock Configuration

The base ADC clock is provided directly by the system clock (SYSCLK). SYSCLK is used to generate the ADC acquisition window. The register ADCCTL2 has a PRESCALE field that determines the ADCCLK. ADCCLK is used to clock the converter, and is only active during the conversion phase. At all other times, including during the sample-and-hold window, the ADCCLK signal is gated off.

In 16-bit mode, the core requires approximately 29.5 ADCCLK cycles to process a voltage into a conversion result, while in 12-bit mode, this process requires approximately 10.5 ADCCLK cycles. The choice of resolution also determines the necessary duration of the acquisition window, see Section 24.14.2.

Note

To determine an appropriate value for ADCCTL2.PRESCALE, see the device data sheet to determine the maximum SYSCLK and ADCCLK frequency.

24.2.2 Resolution

The resolution of the ADC determines how finely the analog range is quantized into digital values. Each ADC module supports 12-bit resolution, with configurable 16-bit resolution available on ADC-A and ADC-B. Refer to the device data sheet for a description of available ADC modules and supported resolutions.

The resolution can be configured by using either the AdcSetMode() or ADC_setMode() functions, depending on the header files used, provided in C2000ware. These functions make sure that the correct trim is loaded into the ADC trim registers, and must be called at least once after a device reset. Do not configure the resolution by directly writing to the ADCCTL2 register.

The resolution can be changed at any time when the ADC is idle (no active or pending SOCs). No wait time is necessary after changing the resolution before conversions can be initiated. If SOCs are active or pending when the resolution is changed, those SOCs can produce incorrect conversion results.

24.2.3 Voltage Reference

24.2.3.1 External Reference Mode

There are two groups of ADCs. The 16-bit ADCs (ADC-A and ADC-B) form the first group, and the 12-bit ADCs (ADC-C, ADC-D and ADC-E) form the second group. Each group of ADCs shares VREFHI and VREFLO inputs. In external reference mode, these pins are used as a ratiometric reference to determine the ADC conversion input range.

See Section 24.14.6 for information on how to supply the reference voltage.

Note

- On devices with no external VREFLO pin, VREFLO is internally connected to the device analog ground, VSSA.
- See the device data sheet to determine the allowable voltage range for VREFHI and VREFLO.
- The external reference mode requires an external capacitor on the VREFHI pin. See the device data sheet for the specific value required.

24.2.3.2 Internal Reference Mode

In internal reference mode, the device drives a voltage out onto the VREFHI pin. The VREFHI and VREFLO pins then set the ADC conversion range.

The internal reference voltage can be configured to be either 2.5V or 1.65V. When the 1.65V internal reference voltage is selected, the ADC input signal is internally divided by 2 before conversion, which effectively makes the ADC conversion range from VREFLO to 3.3V.

The 1.65V internal reference mode is not supported when the ADC is configured for 16-bit resolution.

Note

The internal reference mode also requires an external capacitor on the VREFHI pin. See the device data sheet for the specific value required.

24.2.3.3 Ganged References

On some packages, the voltage reference pins for multiple ADCs can be combined. In this case, configure the ganged references identically when selecting external versus internal reference mode and for selecting an internal reference voltage range of 3.3V or 2.5V.

For example, if ADC A and ADC B reference pins are combined and the desired reference mode is 2.5V internal reference mode, the following reference configuration code can be run:

```
//ADCA VREFHI and ADCB VREFHI share a pin
//ADCA VREFLO and ADCB VREFLO share a pin
//Both references must be explicitly configured
//Both references must be configured identically
SetVREF(ADC_ADCA, ADC_INTERNAL, ADC_VREF2P5);
SetVREF(ADC_ADCB, ADC_INTERNAL, ADC_VREF2P5);
```

Internal device hardware makes sure multiple references do not drive conflicting voltages onto the same pin. Because of this, references can be configured in any order or over any amount of time.

24.2.3.4 Selecting Reference Mode

The voltage reference mode must be configured by using the ADC_setVREF() function, depending on the header files used, provided in C2000Ware. Using either of these functions makes sure that the correct trim is loaded in the ADC trim registers. This function must be called at least once after a device reset. Do not configure the voltage reference mode by directly writing to the ANAREFCTL register.

24.2.4 Signal Mode

ADC-A and ADC-B support two signal modes: single-ended and differential. For the other ADCs, only single-ended signal mode is supported.

In single-ended mode, the input voltage to the converter is sampled through a single pin (ADCINx), referenced to VREFLO.

In differential signaling mode, the input voltage to the converter is sampled through a pair of input pins, one of which is the positive input (ADCINxP) and the other is the negative input (ADCINxN). The actual input voltage is the difference between the two (ADCINxP – ADCINxN).

Note

- In differential signal mode, VREFLO must be connected to VSSA.
- In differential signal mode, the common mode voltage is V_{CM} = (ADCINxP + ADCINxN)/2

  The data sheet for a particular device places some requirements on how close this voltage needs to be to: (VREFHI + VREFLO)/2

  Note: The above condition is not met by connecting the negative input to VSSA or VREFLO.
- Differential signaling mode is advantageous because noise encountered on both inputs is largely canceled. The effect can be maximized by routing the positive and negative traces for the same differential input as close together as possible and keeping them symmetrical with respect to the signal reference.

The signal mode must be configured by using either the ADC_setMode() or AdcSetMode() functions, depending on the header files used, provided in C2000ware. These functions make sure that the correct trim is loaded into the ADC trim registers. These functions must be called at least once after a device reset. The signal mode must not be configured by writing to the ADCCTL2 register directly.

24.2.5 Expected Conversion Results

Based on a given analog input voltage, the expected digital conversion is given in Table 24-2 and Table 24-3. Fractional values are truncated.

Table 24-2. Analog to 12-bit Digital Formulas

	Analog Input	Digital Result
Single-Ended	when ADCINy ≤ VREFLO	ADCRESULTx = 0
	when VREFLO < ADCINy < VREFHI	$ADCRESULTx = 4096\left(\frac{ADCINy - VREFLO}{VREFHI - VREFLO}\right)$
	when ADCINy ≥ VREFHI	ADCRESULTx = 4095
Differential	when ADCINyP - ADCINyN ≤ -(VREFHI-VREFLO)	ADCRESULTx = 0
	when -(VREFHI-VREFLO) < ADCINyP - ADCINyN ≤ (VREFHI-VREFLO)	$ADCRESULTx = 4096\left(\frac{ADCINyP - ADCINyN + VREFHI - VREFLO}{2(VREFHI - VREFLO)}\right)$
	when ADCINyP - ADCINyN ≥ (VREFHI-VREFLO)	ADCRESULTx = 4095

Table 24-3. Analog to 16-bit Digital Formulas

	Analog Input	Digital Result
Single-Ended	when ADCINy ≤ VREFLO	ADCRESULTx=0
	when VREFLO < ADCINy < VREFHI	$ADCRESULTx = 65536\left(\frac{ADCINy - VREFLO}{VREFHI - VREFLO}\right)$
	when ADCINy ≥VREFHI	ADCRESULTx=65535

Table 24-3. Analog to 16-bit Digital Formulas (continued)

	Analog Input	Digital Result
Differential	when ADCINyP - ADCINyN ≤ -(VREFHI-VREFLO)	ADCRESULTx = 0
	when -(VREFHI-VREFLO) < ADCINyP - ADCINyN ≤ (VREFHI-VREFLO)	$ADCRESULTx = 65536\left(\frac{ADCINyP - ADCINyN + VREFHI - VREFLO}{2(VREFHI - VREFLO)}\right)$
	when ADCINyP - ADCINyN ≥ (VREFHI-VREFLO)	ADCRESULTx = 65535

24.2.6 Interpreting Conversion Results

Based on a given ADC conversion result, the corresponding analog input is given in Table 24-4 and Table 24-5. This corresponds to the center of the possible range of analog voltages that can produce this conversion result.

Table 24-4. 12-Bit Digital-to-Analog Formulas

	Digital Value	Analog Equivalent	
Single-Ended	when ADCRESULTy = 0	$\text{ADCINx} \leq \text{VREFLO}$	(2)
	when 0 < ADCRESULTy < 4095	$\text{ADCINx} = (\text{VREFHI} - \text{VREFLO})\left(\frac{\text{ADCRESULTy}}{4096}\right) + \text{VREFLO}$	(3)
	when ADCRESULTy = 4095	$\text{ADCINx} \geq \text{VREFHI}$	(4)
Differential	when ADCRESULTy = 0	$ADCINxP - ADCINxN \leq (VREFHI - VREFLO)$	(5)
	when 0 < ADCRESULTy < 4095	$ADCINxP - ADCINxN = (VREFHI - VREFLO)\left(\frac{ADCRESULTy}{2048} - 1\right)$	(6)
	when ADCRESULTy = 4095	$ADCINxP - ADCINxN \geq (VREFHI - VREFLO)$	(7)

Table 24-5. 16-Bit Digital-to-Analog Formulas

	Digital Value	Analog Equivalent	
Single-Ended	when ADCRESULTy = 0	$\text{ADCINx} \leq \text{VREFLO}$	(8)
	when 0 < ADCRESULTy < 65535	$\text{ADCINx} = (\text{VREFHI} - \text{VREFLO})\left(\frac{\text{ADCRESULTy}}{65536}\right) + \text{VREFLO}$	(9)
	when ADCRESULTy = 65535	$\text{ADCINx} \geq \text{VREFHI}$	(10)
Differential	when ADCRESULTy = 0	$ADCINxP - ADCINxN \leq (VREFHI - VREFLO)$	(11)
	when 0 < ADCRESULTy < 65535	$ADCINxP - ADCINxN = (VREFHI - VREFLO)\left(\frac{ADCRESULTy}{32768} - 1\right)$	(12)
	when ADCRESULTy = 65535	$ADCINxP - ADCINxN \geq (VREFHI - VREFLO)$	(13)

24.3 SOC Principle of Operation

The ADC triggering and conversion sequencing is accomplished through configurable start-of-conversions (SOCs). Each SOC is a configuration set defining the single conversion of a single channel. In that set, there are three configurations: the trigger source that starts the conversion, the channel to convert, and the acquisition (sample) window duration. Upon receiving the trigger configured for a SOC, the wrapper makes sure that the specified channel is captured using the specified acquisition window duration.

Multiple SOCs can be configured for the same trigger, channel, and acquisition window as desired. Configuring multiple SOCs to use the same trigger allows the trigger to generate a sequence of conversions. Configuring multiple SOCs to use the same trigger and channel allows for oversampling. Oversampling can also be achieved using a single trigger source by configuring the trigger repeater module. See Section 24.3.2.2 for more information.

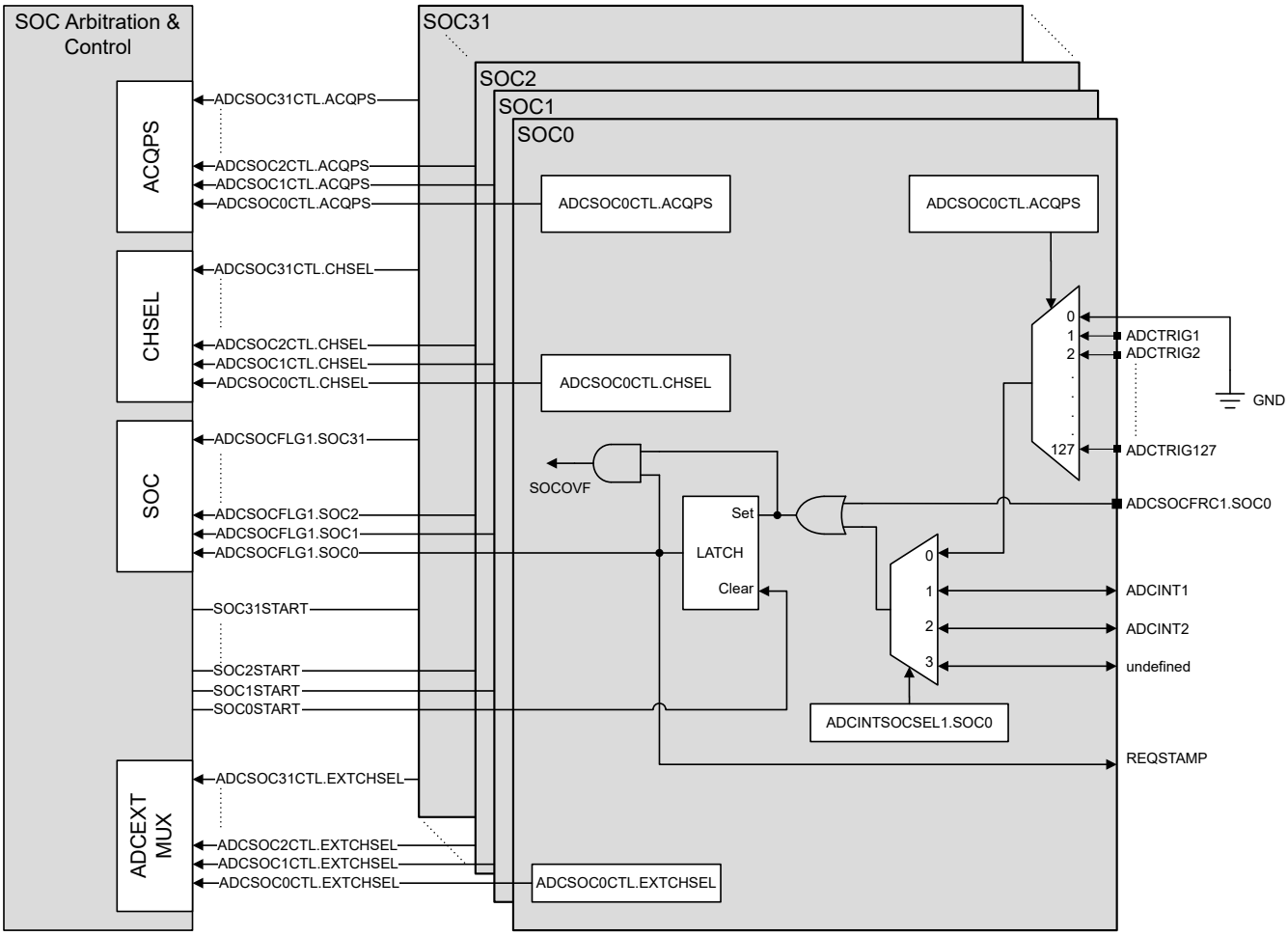


Figure 24-2. SOC Block Diagram

24.3.1 SOC Configuration

Each SOC has a configuration register, ADCSOCxCTL. Within this register, SOCx can be configured for trigger source, channel to convert, optional external channel mux selection, and acquisition (sample) window duration.

24.3.2 Trigger Operation

Each SOC can be configured to start on one of many input triggers. The primary trigger select for SOCx is in the ADCSOCxCTL.TRIGSEL register, which can select between:

- Disabled (software only)
- CPU Timers 0/1/2 (from each C29 core present)
- GPIO: Input X-Bar INPUT5
- ADCSOCA or ADCSOCB from each ePWM module
- eCAP events
- Either of the two trigger repeater blocks. This can be used to achieve oversampling, undersampling, or to apply a trigger delay.
- A global synchronous software trigger. This is achieved by configuring the ADCSOCFRCGBSEL and ADCSOCFRCGB analog subsystem registers.

In addition, each SOC can also be triggered when the ADCINT1 flag or ADCINT2 flag is set. This is achieved by configuring the ADCINTSOCSEL1 register (for SOC0 to SOC15) or the ADCINTSOCSEL2 register (for SOC16 and higher where applicable). This is useful for creating continuous conversions.

Table 24-6. ADC SOC Trigger Selection

Index	Signal
0	ADC_SOFTWARE_TRIGGER
1	CPU1_TINT0
2	CPU1_TINT1
3	CPU1_TINT2
4	INPUTXBAR5
5	EPWM1_ADCSOCA
6	EPWM1_ADCSOCB
7	EPWM2_ADCSOCA
8	EPWM2_ADCSOCB
9	EPWM3_ADCSOCA
10	EPWM3_ADCSOCB
11	EPWM4_ADCSOCA
12	EPWM4_ADCSOCB
13	EPWM5_ADCSOCA
14	EPWM5_ADCSOCB
15	EPWM6_ADCSOCA
16	EPWM6_ADCSOCB
17	EPWM7_ADCSOCA
18	EPWM7_ADCSOCB
19	EPWM8_ADCSOCA
20	EPWM8_ADCSOCB
21	EPWM9_ADCSOCA
22	EPWM9_ADCSOCB

Table 24-6. ADC SOC Trigger Selection (continued)

Index	Signal
23	EPWM10_ADCSOCA
24	EPWM10_ADCSOCB
25	EPWM11_ADCSOCA
26	EPWM11_ADCSOCB
27	EPWM12_ADCSOCA
28	EPWM12_ADCSOCB
29-39	Reserved
40	ADC_REP1TRIG
41	ADC_REP2TRIG
42-71	Reserved
72	CLB1_OUT27
73	CLB2_OUT27
74	CLB3_OUT27
75	CLB4_OUT27
76	CLB5_OUT27
77	CLB6_OUT27
78	Reserved
79	Reserved
80	ECAP1_SOC
81	ECAP2_SOC
82	ECAP3_SOC
83	ECAP4_SOC
84	ECAP5_SOC
85	ECAP6_SOC
86-87	Reserved
88	EPWM13_ADCSOCA
89	EPWM13_ADCSOCB
90	EPWM14_ADCSOCA
91	EPWM14_ADCSOCB
92	EPWM15_ADCSOCA
93	EPWM15_ADCSOCB
94	EPWM16_ADCSOCA
95	EPWM16_ADCSOCB
96	EPWM17_ADCSOCA
97	EPWM17_ADCSOCB
98	EPWM18_ADCSOCA
99	EPWM18_ADCSOCB
100	CPU2_TINT0

Table 24-6. ADC SOC Trigger Selection (continued)

Index	Signal
101	CPU2_TINT1
102	CPU2_TINT2
103	CPU3_TINT0
104	CPU3_TINT1
105	CPU3_TINT2
106-127	Reserved

24.3.2.1 Global Software Trigger

This ADC supports synchronous global software triggers. Synchronous global triggers allow the application to trigger SOCs on multiple ADC instances that are exactly simultaneous in time. To generate a global software trigger, configure the analog subsystem register ADCSOCFRCGBSEL to select the ADC instances to be triggered, then write to ADCSOCFRCGB to trigger the desired SOCs simultaneously on each ADC.

For example, to trigger SOC0, SOC1, and SOC2 on ADCA and ADCC:

1. Set ADCSOCFRCGBSEL.ADCA = 1 and ADCSOCFRCGBSEL.ADCC = 1 by writing 0x5 to the ADCSOCFRCGBSEL register.
2. Trigger SOCs 0, 1, and 2 by writing 0x7 to the ADCSOCFRCGB register.

24.3.2.2 Trigger Repeaters

Each ADC instance contains two trigger repeater modules. These modules can select any of the regular ADC triggers that are selectable by the ADCSOCxCTL.TRIGGER register, and generate a number of repeat pulses as configured in the REPxN.NSEL register. Figure 24-3 shows a functional block diagram of the ADC trigger repeater module.

Each repeater module can apply four types of trigger modifications:

- Oversampling mode
- Undersampling mode
- Phase delay
- Re-trigger spread

Each of these trigger modification features is explained in detail in the following sections.

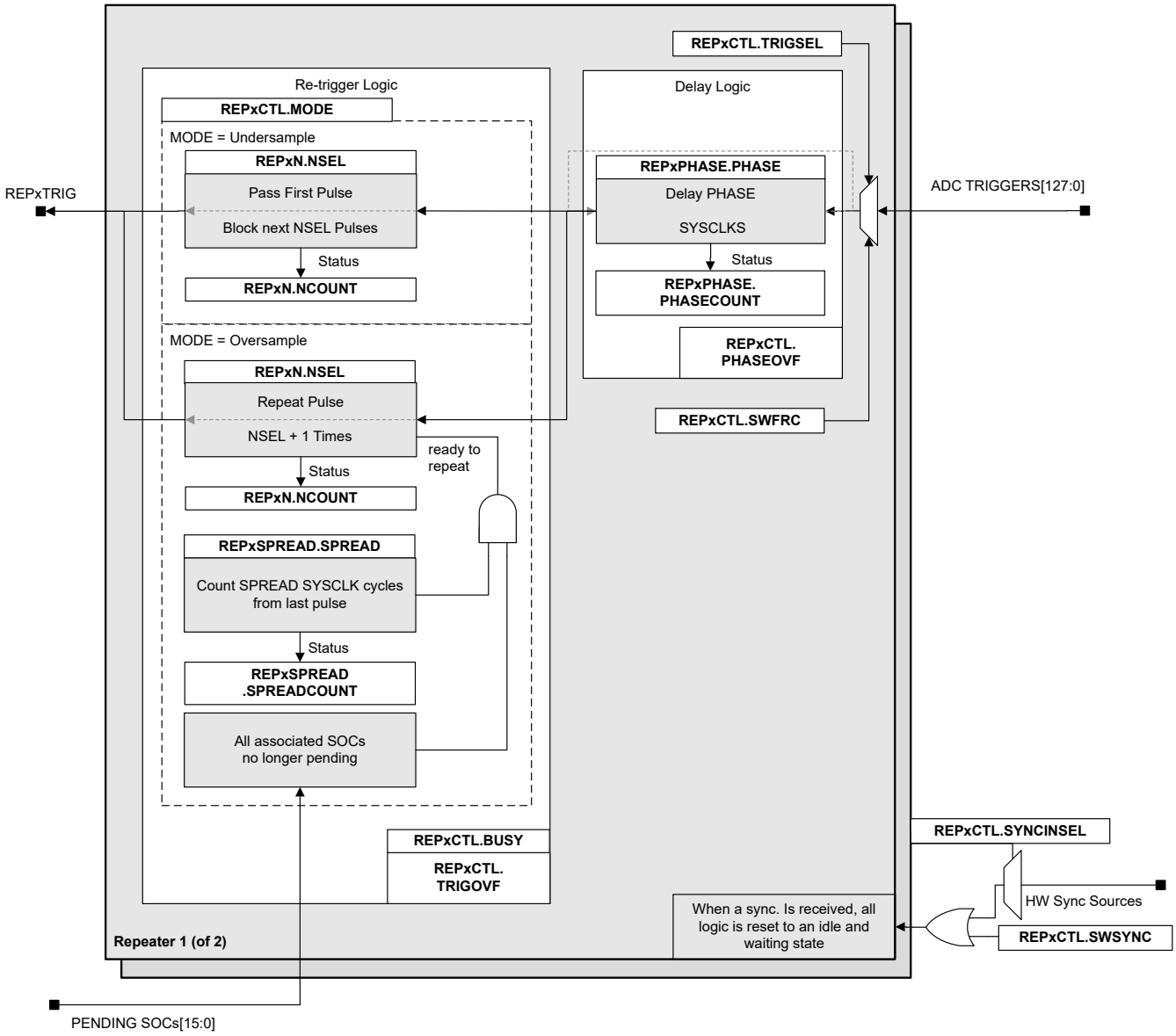
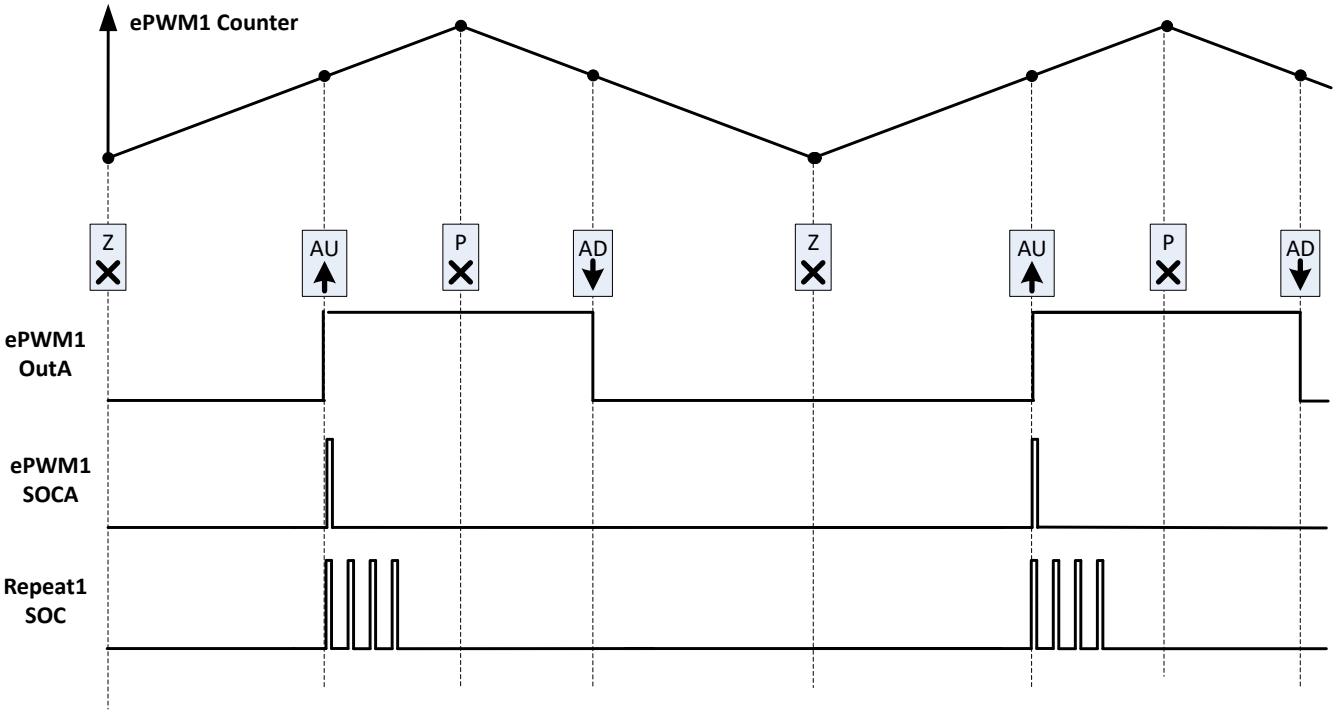


Figure 24-3. ADC Trigger Repeater Block Diagram

24.3.2.2.1 Oversampling Mode

In this mode, the repeater module passes the initial trigger through to the output. As soon as all SOCs configured to receive the trigger are in progress or completed, the repeater issues the trigger again. The process repeats until the configured number of trigger pulses (NSEL + 1) have been issued.

This mode allows the application to easily perform multiple back-to-back samples from a single trigger pulse. When used in conjunction with the aggregation options in the post-processing block, this mode enables oversampling, averaging, or peak detection. Figure 24-4 shows an example of oversampling SOCs generated from a single ePWM trigger.



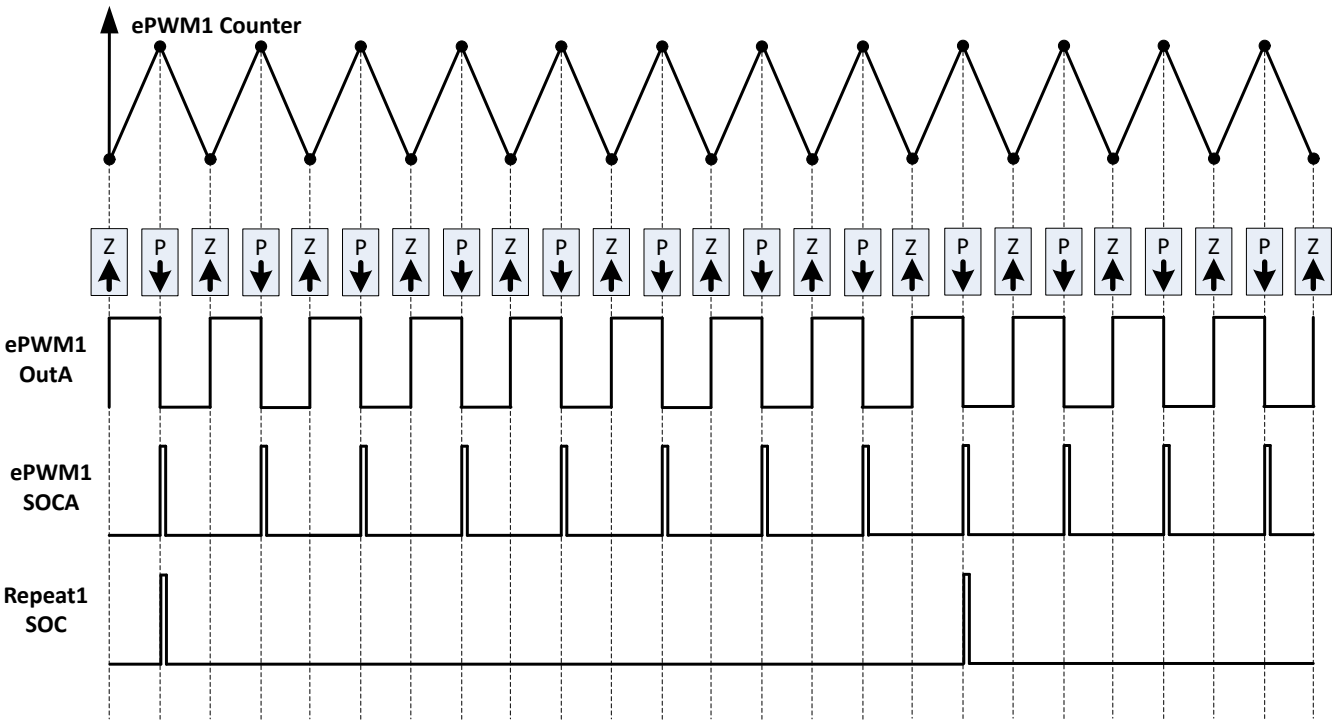
TRIGGER = ePWM SOCA, NSEL = 3, PHASE = 0, MODE = Oversampling, SPREAD = 0

Figure 24-4. Oversampled ADC Trigger Example

24.3.2.2.2 Undersampling Mode

In this mode, the repeater module passes the initial trigger through to the output, and then blocks subsequent triggers until the configured number of trigger pulses (NSEL + 1) arrive. The result is that only 1 in every (NSEL + 1) pulses passes through to the output. Figure 24-5 shows an example of undersampled SOCs from multiple ePWM triggers.

This mode enables the application to scale down the trigger frequency for one or more SOCs. This is useful for charge-sharing input drivers which have increased error with higher sampling frequencies.



TRIGGER = ePWM SOCA, NSEL = 7, PHASE = 0, MODE = Undersampling, SPREAD = (don't care)

Figure 24-5. Undersampled ADC Trigger Example

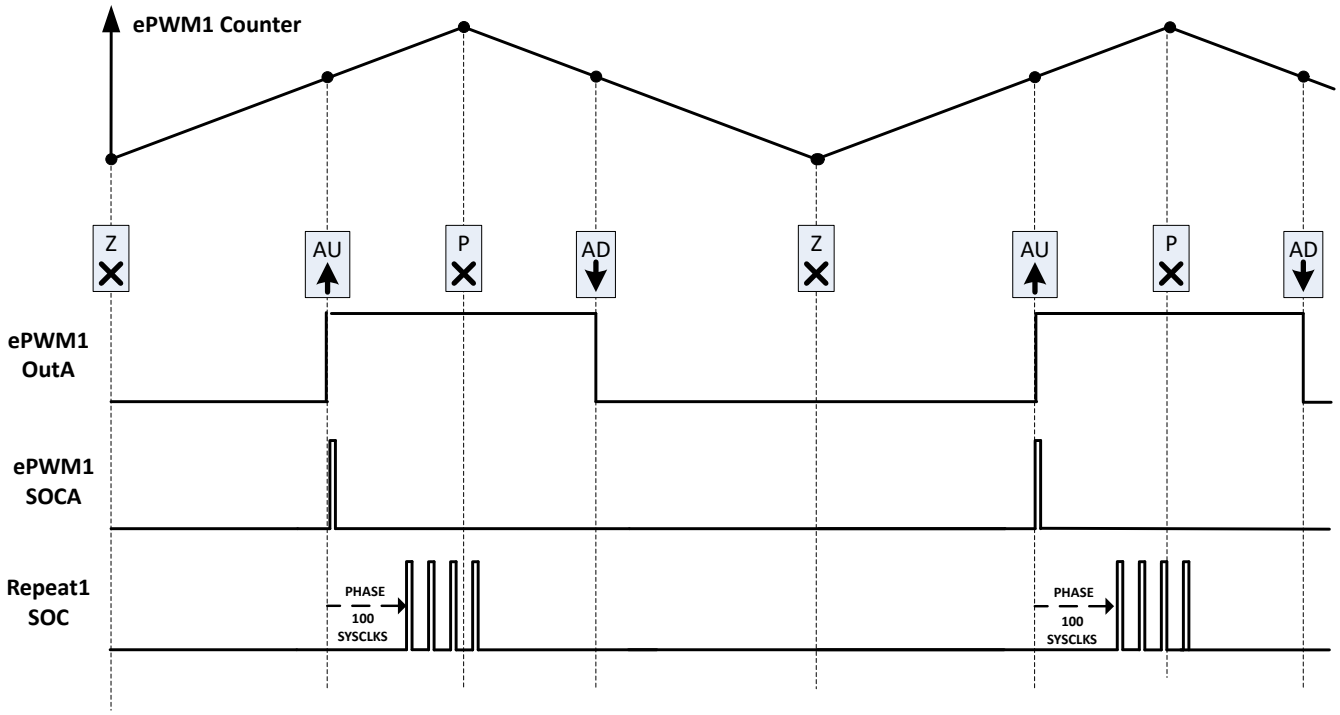
Note

Oversampling and undersampling modes are mutually exclusive for each repeater module. However, multiple repeater modules can (and are intended to) be used in different modes concurrently.

24.3.2.2.3 Trigger Phase Delay

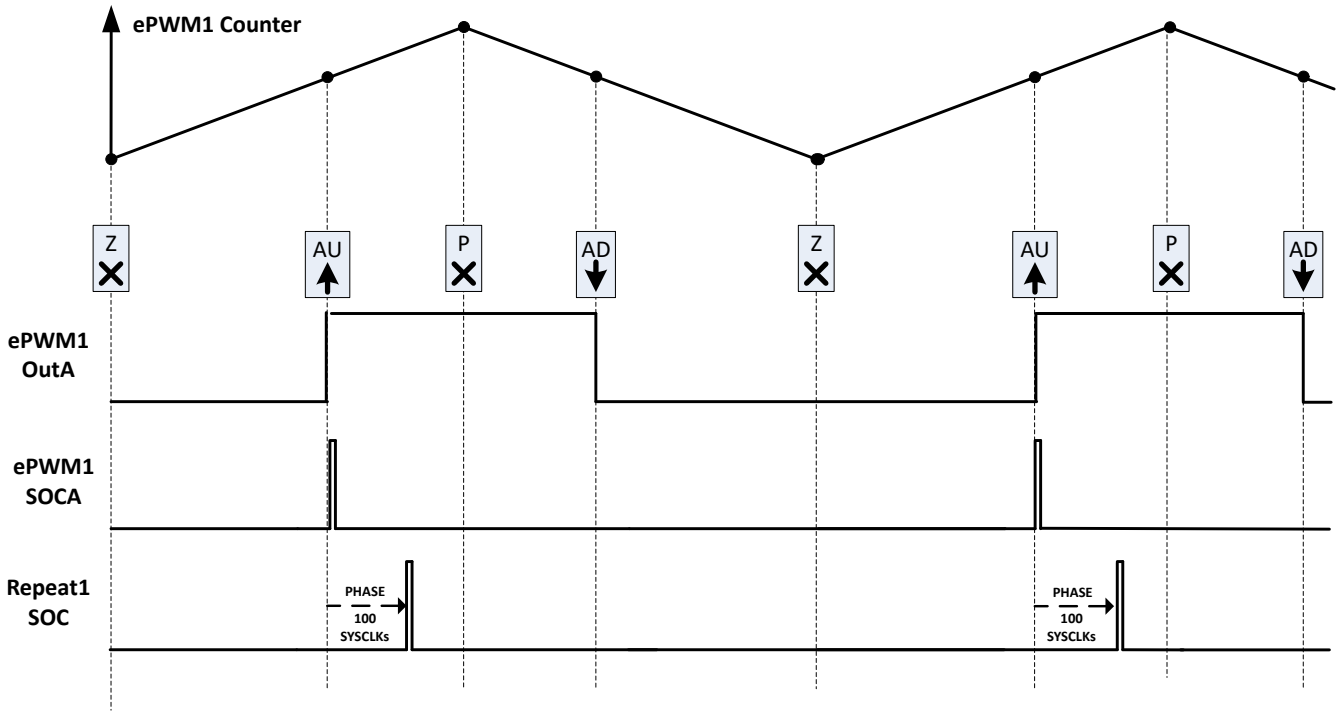
The repeater module can delay the initial trigger by a specified number of SYSCLK cycles. This feature can be used in combination with oversampling or undersampling modes, or as a standalone delay by setting NSEL = 0. The phase delay does not affect the timing between subsequent repeated oversampled triggers—the phase delay only delays the initial trigger. When PHASE = 0, the initial trigger arrives at the same time as an unmodified trigger. Figure 24-6 shows an example of phase delay combined with oversampling. Figure 24-7 shows an example of a standalone phase delay with a single SOC trigger.

Phase delay enables the application to tie the trigger start point to an ePWM event while allowing for a necessary sampling delay (for example, settling time). In addition, when phase delay is combined with oversampling functionality, a single trigger can generate an interleaved burst of conversions across multiple ADCs. To achieve this, set PHASE in increments of (t_{sample}/n_interleaved_ADCs). Figure 24-8 shows an example of interleaving 12 samples across 3 ADCs.



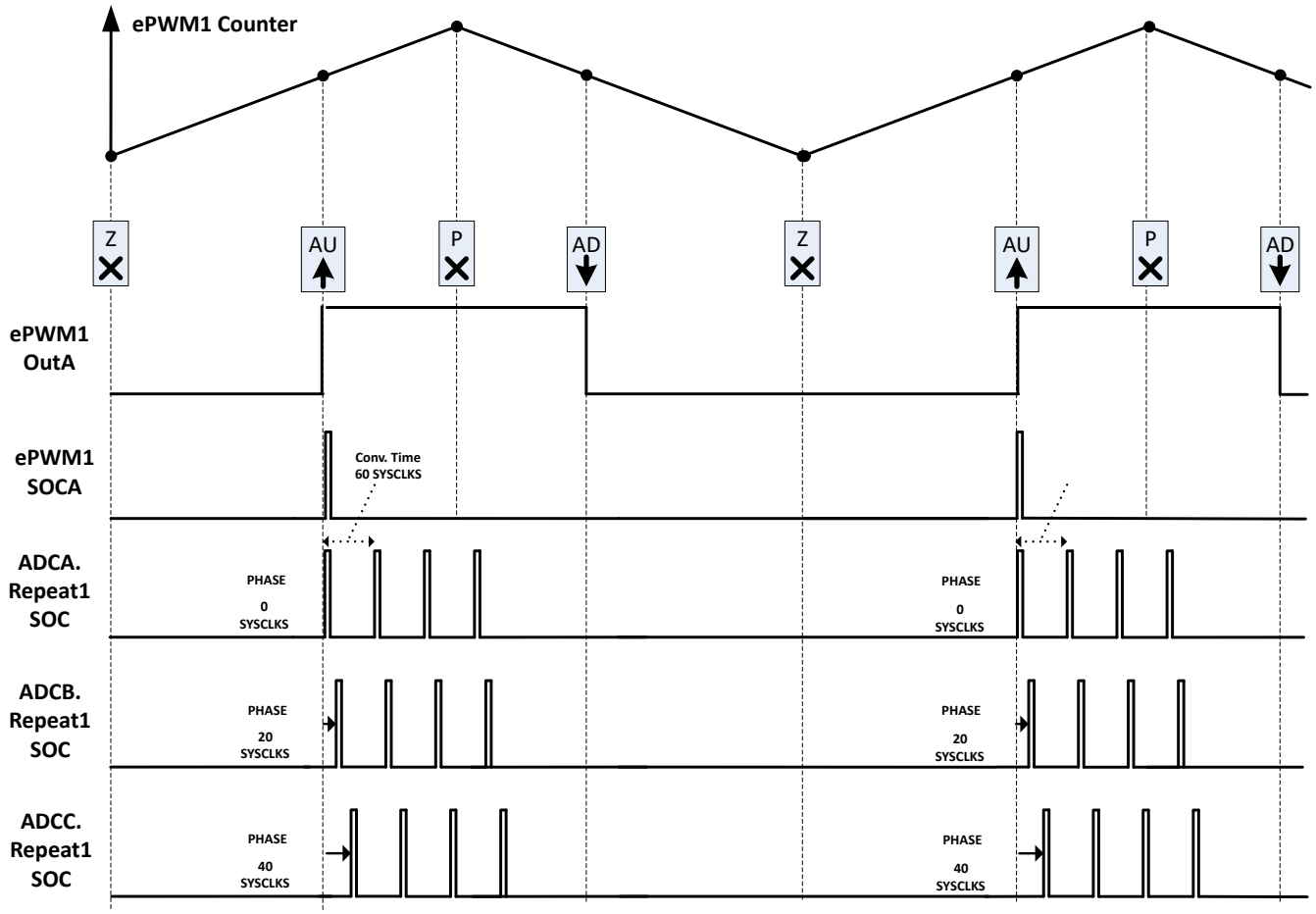
TRIGGER = ePWM SOCA, NSEL = 3, PHASE = 100, MODE = Oversampling, SPREAD = 0

Figure 24-6. Oversampled ADC Trigger Example with Phase Delay



TRIGGER = ePWM SOCA, NSEL = 0, PHASE = 100, MODE = (either), SPREAD = (don't care)

Figure 24-7. ADC Trigger Example with Phase Delay



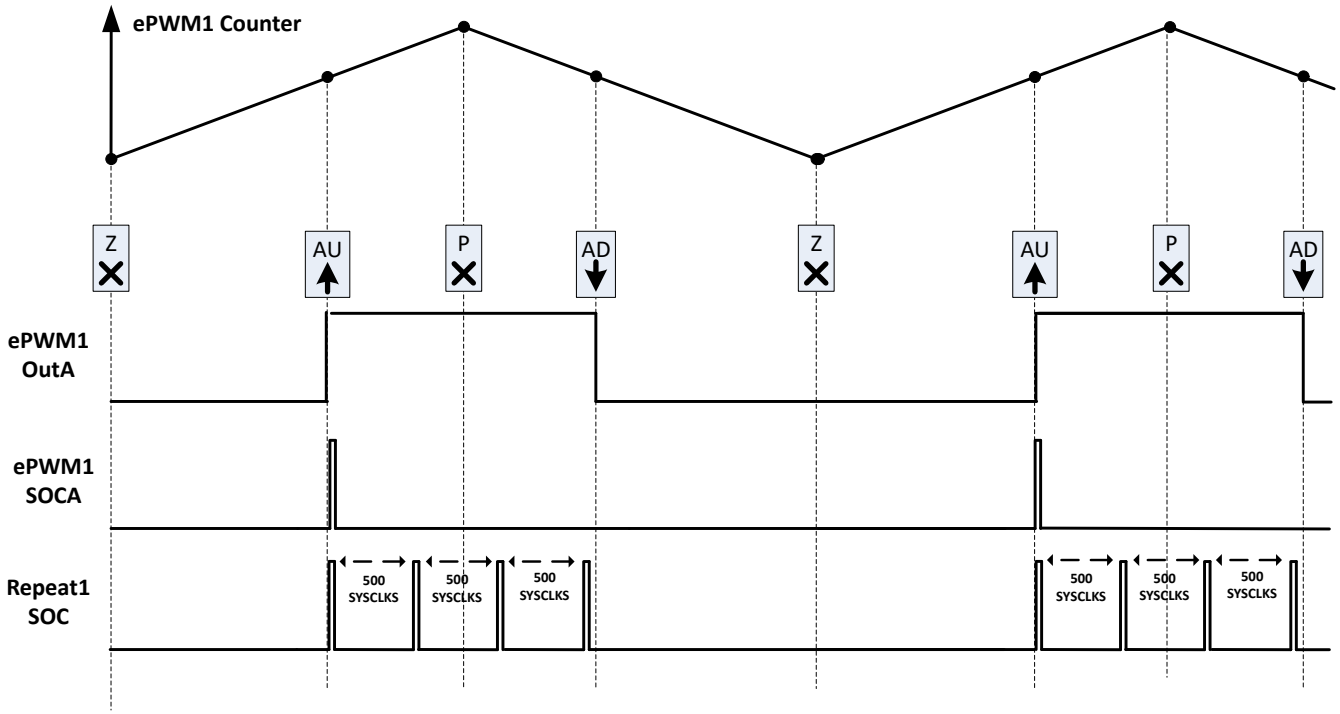
TRIGGER = ePWM SOCA, NSEL = 3, MODE = Oversampling, PHASE = (varies per ADC), SPREAD = 0

Figure 24-8. ADC Interleaved Trigger Example (12 Samples Across 3 ADCs)

24.3.2.2.4 Re-trigger Spread

If additional time between samples is desired, the application can configure SPREAD equal to the number of SYSCLK cycles desired between samples. Figure 24-9 shows an example of oversampling from an ePWM trigger with a 500-cycle spread between samples.

- By default, SPREAD = 0, and samples are re-triggered as soon as all associated SOCs are no longer pending.
- If SPREAD is set to a value smaller than the time needed for the associated SOCs to complete, then the ADC performs the triggered conversions back-to-back, and SPREAD is effectively 0.
- SPREAD has no effect in undersampling mode, or when NSEL = 0.



TRIGGER = ePWM SOCA, NSEL = 3, MODE = Oversampling, PHASE = 0, SPREAD = 500

Figure 24-9. ADC Repeated Trigger Example with Sample Spread

24.3.2.2.5 Trigger Repeater Configuration

To configure ADC oversampling or undersampling using the trigger repeater module, follow this procedure:

1. Set up the SOC by writing to ADCSOCxCTL. Specify one of the two repeater modules (REP1TRIG or REP2TRIG) as the trigger source.
2. Configure the repeater module by writing to the REPxCTL register:
   a. Configure oversampling or undersampling mode using the MODE bit.
   b. Specify the desired SOC trigger source in the TRIGGER field.
   c. If desired, configure a sync source for the repeater module in the SYNCINSEL field. A sync event resets all repeater registers to a ready and waiting state, while preserving NSEL, PHASE and MODE. A software-initiated sync is also possible by writing 1 to the SWSYNC bit.
   d. If desired, clear any previously set phase and trigger overflow flags by writing to the PHASEOVF and TRIGGEROVF bits.
3. Configure the trigger repeat count by writing to the REPxN.NSEL register. The repeater module supports up to 128 repeats for each trigger.
4. Configure the repeater phase delay by writing to the REPxPHASE.PHASE register.
5. To configure a re-trigger spread delay in oversampling mode, write the desired delay value in SYSCLK cycles to the REPxSPREAD.SPREAD register.
6. Configure the PPBLIMIT register. This register defines how many samples the post-processing block accumulates before loading the partial sum value in ADCPPBxPSUM into ADCPPBXSUM.
7. The post-processing block (PPB) and trigger repeater module have independent sync source configurations. To configure the PPB sync source, write to the ADCPPBxCONFIG2 register. For more information on how to configure the ADC post-processing block, see Section 24.8.

Note

When NSEL = 0, the repeater module essentially acts as a pass-through for SOC triggers, but is still useful for applying phase delay. SOC triggers are passed through in both oversampling mode and undersampling mode, even if there are still pending SOCs. In this scenario, the ADC sets a trigger overflow flag for the individual SOC (ADCSOCOVF1.SOCx), not the repeater module. When oversampling with NSEL > 0, the ADC sets the oversampled trigger overflow flag (REPxCTL.TRIGGEROVF) if a trigger arrives while there are pending SOCs.

When NSEL = 0, the repeater module does not set the REPxCTL.MODULEBUSY indicator. In this scenario, the application must make sure that all associated SOC flags have completed before enabling oversampling or undersampling mode by setting NSEL > 0.

24.3.2.2.5.1 Register Shadow Updates

To avoid latency or processing delays between triggers, the application can write updated values to the NSEL, PHASE and SPREAD registers while the repeater module is still actively working. When a new SOC trigger is received, these values are loaded into the NCOUNT, PHASECOUNT and SPREADCOUNT registers, which then count down to zero as each SOC is triggered by the repeater module.

In addition, the application can change the repeater module's oversampling or undersampling mode by writing to the REPxCTL.MODE register while the repeater is actively working, without affecting the current operation. The repeater module loads the value of REPxCTL.MODE into REPxCTL.ACTIVEMODE when a new SOC trigger is received.

24.3.2.2.6 Re-Trigger Logic

The repeater module determines when to re-trigger based on the values of ADCSOCxCTL.TRIGSEL and the SOC flags in ADCSOCFLG1. A repeat trigger is issued when all SOCs configured to be triggered by the repeater module instance are no longer pending. Figure 24-10 describes the trigger repeat logic. Because the SOC pending flag goes low at the end of the sample and hold phase, the module has plenty of time to re-trigger conversions without introducing any latency between repeat conversions.

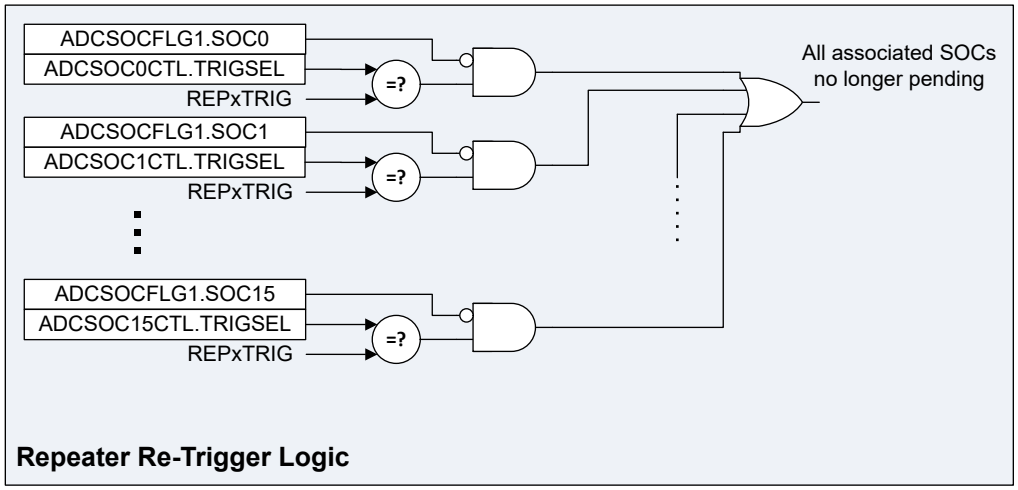


Figure 24-10. Trigger Repeater Repeat Logic

Re-triggering in Burst Mode

If the ADC is in burst mode, and the repeater is selected as the BURSTTRIG source, then the repeater fires a re-trigger pulse whenever there are no high-priority associated SOCs pending, and there are no round-robin SOCs pending.

24.3.2.2.7 Multi-Path Triggering Behavior

With the trigger repeater modules, it is possible to have one trigger source take multiple paths to set SOCs in various ways. For example, ePWM1 can directly trigger SOC3 and SOC4, while one repeater block uses ePWM1 to generate oversampling triggers on SOCs 0-2, and the second repeater block generates undersampled triggers to SOC5. Assuming all SOCs are configured for round-robin priority and the various triggers all arrive in the same cycle, the conversion order is SOC0 to SOC5 in increasing order; this is then followed by the oversampled conversions on SOC0-SOC2.

24.3.3 ADC Acquisition (Sample and Hold) Window

External signal sources vary in the ability to drive an analog signal quickly and effectively. To achieve rated resolution, the signal source needs to charge the sampling capacitor in the ADC core to within 0.5 LSBs of the signal voltage. The acquisition window is the amount of time the sampling capacitor is allowed to charge and is configurable for SOCx by the ADCSOCxCTL.ACQPS register.

ACQPS is a 9-bit register that can be set to a value between 0 and 511, resulting in an acquisition window duration of:

Acquisition window = (ACQPS + 1) * (System Clock (SYSCLK) cycle time)

- The acquisition window duration is based on the System Clock (SYSCLK), not the ADC clock (ADCCLK).
- The selected acquisition window duration must be at least as long as one ADCCLK cycle.
- The data sheet specifies a minimum acquisition window duration (in nanoseconds). The user is responsible for selecting an acquisition window duration that meets this requirement.

24.3.4 ADC Input Models

For single-ended operation, the ADC input characteristics for values in the single-ended input model (see Figure 24-11) can be found in the device data sheet.

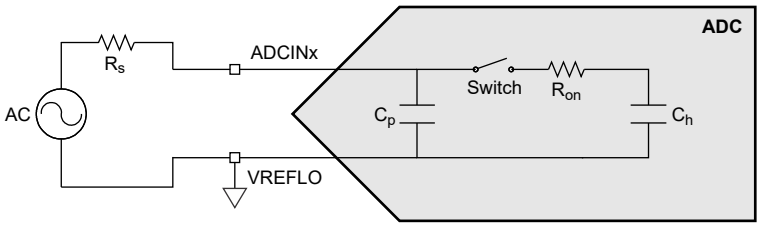


Figure 24-11. Single-Ended Input Model

For differential operation, the ADC input characteristics for values in the differential input model (see Figure 24-12) can be found in the device data sheet.

These input models must be used along with actual signal source impedance to determine the acquisition window duration. See Section 24.14.2 for more information.

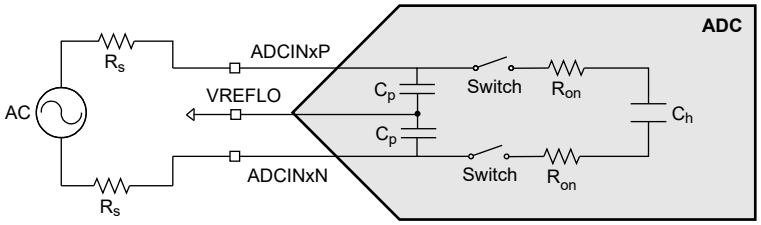


Figure 24-12. Differential Input Model

24.3.5 Channel Selection

Each SOC can be configured to convert any of the ADC channels. This behavior is selected for SOCx by the ADCSOCxCTL.CHSEL register. Depending on the signal mode, the selection is different. For single-ended signal mode, the value in CHSEL selects a single pin as the input. For differential signal mode, the value in CHSEL selects an even-odd pin pair to be the positive and negative inputs. This is summarized in Table 24-7.

Table 24-7. Channel Selection of Input Pins

Input Mode	CHSEL	Input
Single-Ended	0	ADCIN0
	1	ADCIN1
	2	ADCIN2
	3	ADCIN3
	4	ADCIN4
	5	ADCIN5
	6	ADCIN6
	7	ADCIN7
	8	ADCIN8
	9	ADCIN9
	10	ADCIN10
	11	ADCIN11
	12	ADCIN12
	13	ADCIN13
	14	ADCIN14
	15	ADCIN15
	16	ADCIN16
	17	ADCIN17
	18	ADCIN18
	19	ADCIN19
	20	ADCIN20
	21	ADCIN21
	22	ADCIN22
	23	ADCIN23
	24	ADCIN24
	25	ADCIN25
	26	ADCIN26
	27	ADCIN27
	28	ADCIN28
	29	ADCIN29
	30	ADCIN30
	31	ADCIN31

Table 24-7. Channel Selection of Input Pins (continued)

Input Mode	CHSEL	Input	
	CHSEL	Positive Input	Negative Input
Differential	0 or 1	ADCIN0	ADCIN1
	2 or 3	ADCIN2	ADCIN3
	4 or 5	ADCIN4	ADCIN5
	6 or 7	ADCIN6	ADCIN7
	8 or 9	ADCIN8	ADCIN9
	10 or 11	ADCIN10	ADCIN11
	12 or 13	ADCIN12	ADCIN13
	14 or 15	ADCIN14	ADCIN15
	16 or 17	ADCIN16	ADCIN17
	18 or 19	ADCIN18	ADCIN19
	20 or 21	ADCIN20	ADCIN21
	22 or 23	ADCIN22	ADCIN23
	24 or 25	ADCIN24	ADCIN25
	26 or 27	ADCIN26	ADCIN27
	28 or 29	ADCIN28	ADCIN29
	30 or 31	ADCIN30	ADCIN31

24.3.5.1 External Channel Selection

The ADCSOCxCTRL.EXTCHSEL field for each SOC can be used to automatically control an external mux with digital output pins ADCxEXTMUX[3:0]. This functionality enables the application to add additional ADC channels using an external mux, with minimal software overhead. The ADCxEXTMUX[3:0] outputs can be mapped to GPIO pins by configuring the GPIO output crossbar, or configuring the GPIO mux accordingly. The EXTCHSEL field supports up to 4-bit muxes, but fewer mux selection output pins can be configured if desired.

To select a specific channel on the external mux, configure ADCSOCxCTRL.CHSEL to select the ADC pin that is connected to the mux output, and configure ADCSOCxCTRL.EXTCHSEL to select the desired mux input channel. There are a variety of potential mux topologies possible. A basic example can be a single external mux connected to a single ADC input channel. This setup is illustrated in Figure 24-13.

- To select ADCxIN0.4, the user configures the SOC with CHSEL = 0 and EXTCHSEL = 4.
- To select ADCxIN3, the user configures the SOC with CHSEL = 3. The value of EXTCHSEL does not affect the conversion.

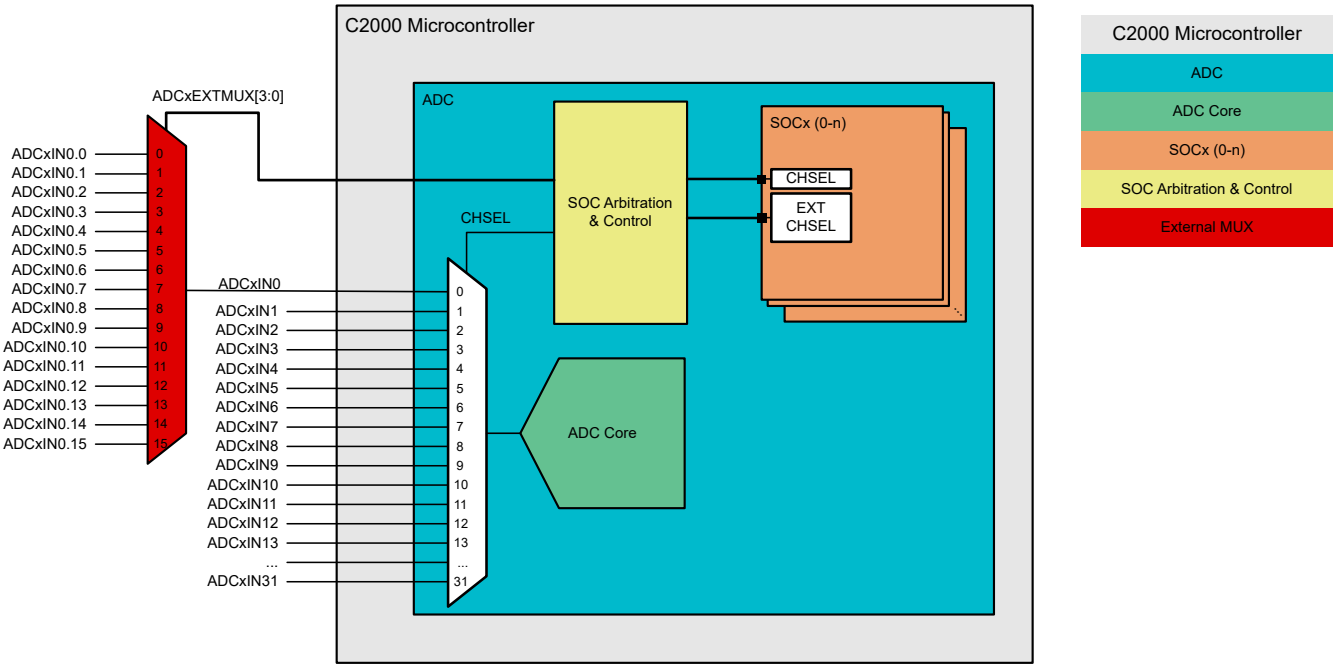


Figure 24-13. ADC with External Input Mux

Another example can be to use multiple external muxes connected to different ADC inputs. This setup is illustrated in Figure 24-14.

- To select ADCxIN0.4, the user configures the SOC with CHSEL = 0 and EXTCHSEL = 4.
- To select ADCxIN3.2, the user configures the SOC with CHSEL = 3 and EXTCHSEL = 2.
- To select ADCxIN5, the user configures the SOC with CHSEL = 5. The value of EXTCHSEL does not affect the conversion.

This scheme saves one digital mux pin, at the expense of using two small muxes instead of a single large mux, and a second ADC input pin.

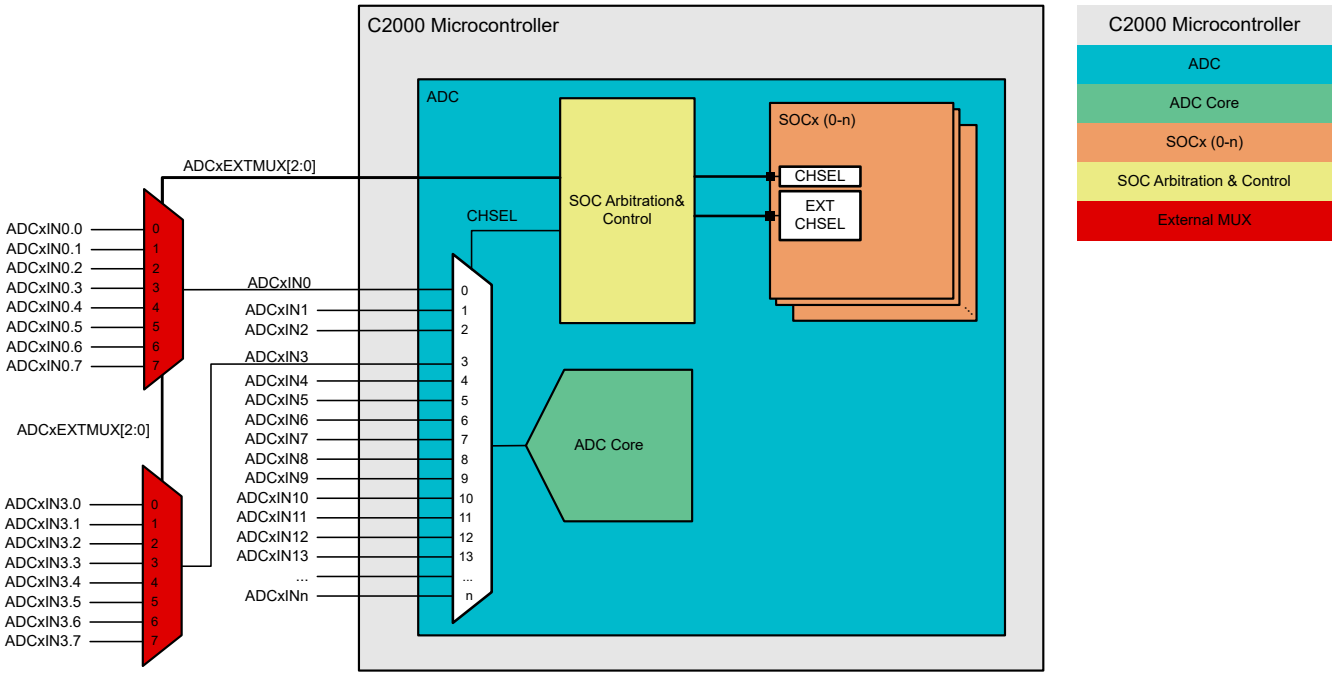


Figure 24-14. ADC with Multiple External Input Muxes and Shared Selection

When using an external channel mux, make sure to comprehend the mux selection and switching delay in the sample/hold time requirement for the SOC. This requirement includes the propagation delay for the output X-BAR (if this is used to configure the mux selection pin), any mux switching delays, and the total resistance and capacitance added to the ADC input network by the external mux device. For more information on calculating the acquisition window size, see Section 24.14.2.

Note

While the external channel selection can technically be used to select up to 16 times the number of total ADC pins (by placing a 16-input mux on each external ADC channel), the system incurs significant added overhead if the total number of channels (internal and external) exceeds 32 – the total number of available SOCs available on the ADC instance. A sensible option is to map the ADCxEXTMUX outputs to ADC pins (AGPIOs). For instance, using one ADC input to sample an external mux output, and 4 inputs for external channel selection, the application effectively gains an additional 11 ADC inputs (from 5 to 16 pins), without using up any additional device pins.

Externally multiplexed ADC channels lose the ability to practically use the analog comparators in the comparator subsystem. However, the digital limit compares in the post-processing block can still be used to generate interrupts and/or PWM trips. For more information, see Section 24.8.

24.3.5.1.1 External Channel Selection Timing

The ADCxEXTMUX output follows two possible timing schemes, depending on the setting of the EXTMUXPRESELECTEN bit in the ADCCTL1 register:

- When EXTMUXPRESELECTEN is set to 0, the ADCxEXTMUX output changes at the beginning of the associated SOC's sample and hold period. The applied external mux setting is maintained until the start of the next SOC sample-and-hold period. This is the default configuration at reset. Examples of SOC timings in this mode are shown in Figure 24-15 and Figure 24-17. When external mux preselect is disabled, make sure to configure the SOC acquisition window duration to account for both external mux settling time and internal channel settling time.
- When EXTMUXPRESELECTEN is set to 1, the ADCxEXTMUX output changes at least one SYSCLK cycle after the end of the sample and hold period. At this point, the ADC sets the external mux selection based on the next highest priority SOC that is pending. If no SOC is pending, then the mux selection is based on the next highest priority SOC, based on the current SOC priority scheme (see Section 24.5 for more information on SOC priority schemes). Examples of SOC timings in this mode are shown in Figure 24-16 and Figure 24-18. Enabling preselect mode enables the application to avoid increasing the acquisition window duration due to external mux switching and settling delays.

Note

When EXTMUXPRESELECTEN is enabled, setting SOC0 as high priority without actually triggering SOC0 conversions is a good way to define an idle value for ADCxEXTMUX. SOC0 always has the highest priority when no SOCs are pending, so the value of ADCSOC0CTL.EXTCHSEL is always pushed onto the mux select pins by default.

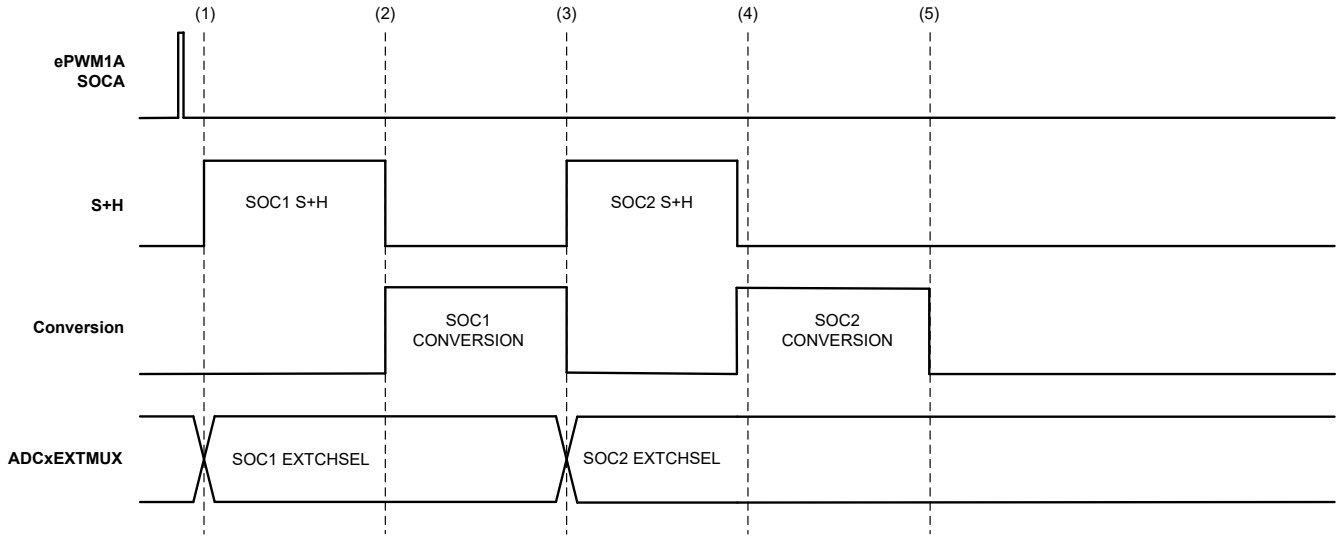


Figure 24-15. ADC External Channel Select Timing Example

In Figure 24-15, the ADC is configured as follows:

- SOC1 and SOC2 are triggered from ePWM1A;
- No high priority SOCs are defined.

The ADC performs the SOC operation sequence in the following order:

1. The initial ePWM1A trigger arrives, setting the SOC1 and SOC2 to pending. SOC1 gains priority, and the sample-and-hold period for SOC1 begins. The ADC pushes the value of ADCSOC1CTL.EXTCHSEL onto the ADCxEXTMUX pins at the same time when the SOC1 sample-and-hold begins.
2. At the end of the sample-and-hold for SOC1, the conversion begins. ADCxEXTMUX remains unchanged until the start of the next SOC sample-and-hold period.

3. In this example case, there are no asynchronous high priority triggers defined, so SOC2 sample-and-hold starts as soon as SOC1 conversion ends. The ADC pushes ADCSOC2CTL.EXTCHSEL onto the ADCxEXTMUX pins.
4. At the end of the sample-and-hold for SOC2, there are no more pending SOCs. SOC2 begins conversion, and ADCxEXTMUX is unchanged.
5. The SOC2 conversion ends. There are no pending SOCs or triggers, so ADCxEXTMUX remains unchanged.

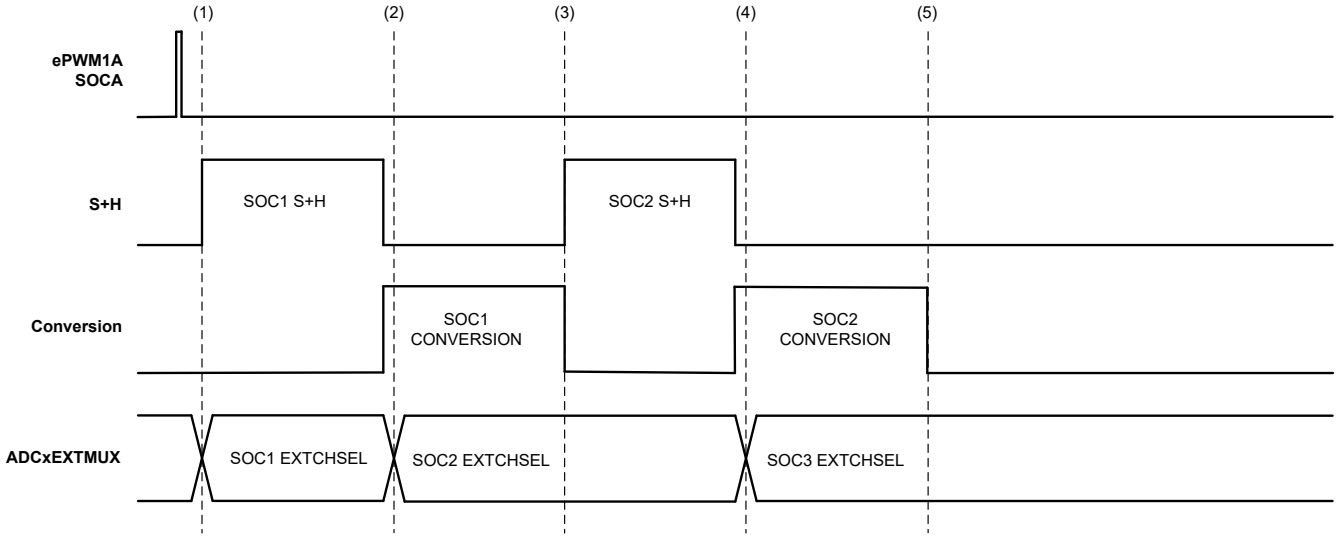


Figure 24-16. ADC External Channel Timing Example in Preselect Mode

In Figure 24-16, the ADC is configured as follows:

- SOC1 and SOC2 are triggered from ePWM1A;
- No high priority SOCs are defined.

The ADC performs the SOC operation sequence in the following order:

1. The initial ePWM1A trigger arrives, setting the SOC1 and SOC2 to pending. SOC1 gains priority, and the sample-and-hold period for SOC1 begins. The ADC pushes the value of ADCSOC1CTL.EXTCHSEL onto the ADCxEXTMUX pins at the same time when the SOC1 sample-and-hold begins.
2. At the end of the sample-and-hold for SOC1, the highest priority SOC that is pending is SOC2, so the ADC pushes the value of ADCSOC2CTL.EXTCHSEL onto the ADCxEXTMUX pins.
3. In this example case, there are no asynchronous high priority triggers defined, so SOC2 sample-and-hold starts as soon as SOC1 conversion ends. The ADC pushes ADCSOC2CTL.EXTCHSEL onto the ADCxEXTMUX pins again, but this is already the current value so there is no change.
4. At the end of the sample-and-hold for SOC2, there are no more pending SOCs. SOC3 has the next highest priority by way of the round-robin pointer, so the ADC pushes the value of ADCSOC3CTL.EXTCHSEL onto ADCxEXTMUX. In this case, the application can set ADCSOC3CTL.EXTCHSEL = ADCSOC1CTL.EXTCHSEL. Although SOC3 is not actually used, this makes sure that the external mux channel is already preselected when the next ePWM1 SOC arrives.
5. The SOC2 conversion ends. There are no pending SOCs or triggers, so ADCxEXTMUX remains unchanged.

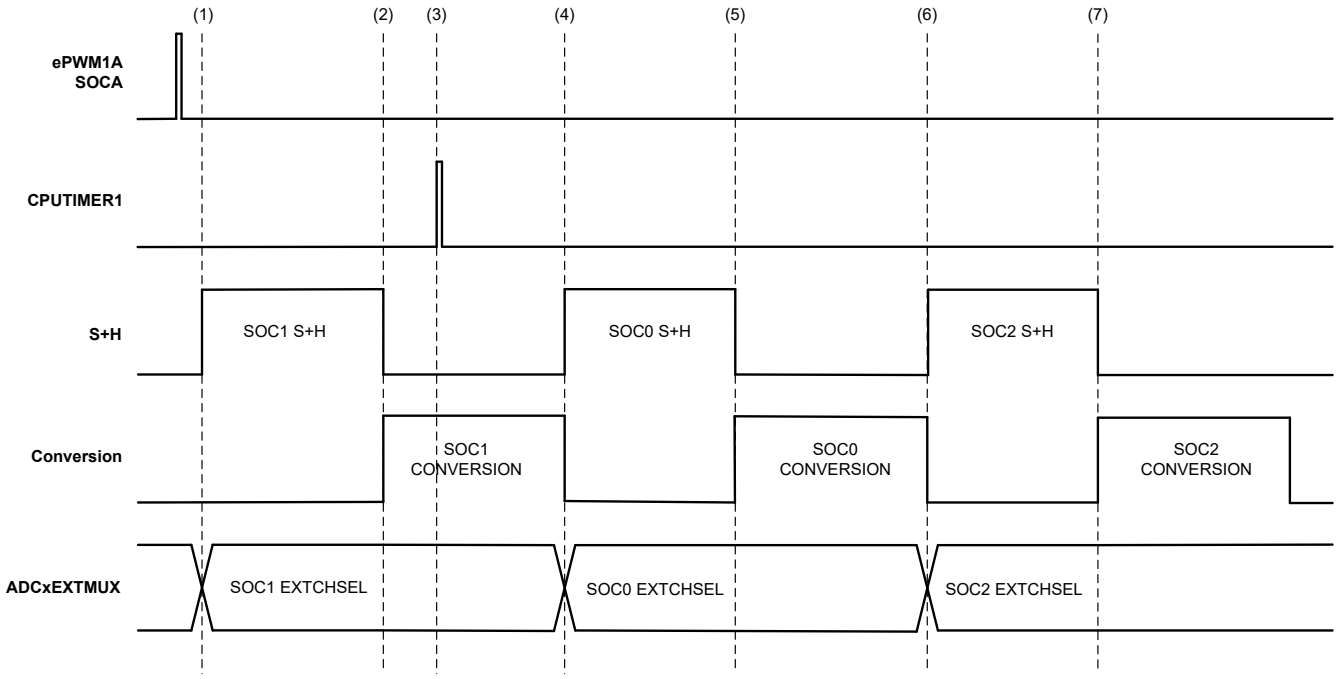


Figure 24-17. ADC External Channel Select Timing Example with Asynchronous Trigger

In Figure 24-17, the ADC is configured as follows:

- SOC1 and SOC2 are triggered from ePWM1A;
- SOC0 is triggered from CPUTIMER1, and has a high priority.

With this configuration, the ADC performs the SOC operation sequence in the following order:

1. The initial ePWM1A trigger arrives, setting the SOC1 and SOC2 flags to pending. SOC1 gains priority, and the SOC1 sample-and-hold period begins. The ADC pushes the value of ADCSOC1CTL.EXTCHSEL onto the ADCxEXTMUX pins at the same time when the SOC1 sample-and-hold begins.
2. At the end of the sample-and-hold for SOC1, the SOC1 conversion begins. ADCxEXTMUX remains unchanged until the start of the next SOC sample-and-hold period.
3. CPUTIMER1 issues a trigger asynchronously, setting the SOC0 flag to pending.
4. Since SOC0 has high priority, SOC0 converts next instead of SOC2. The ADC pushes ADCSOC0CTL.EXTCHSEL onto ADCxEXTMUX when the sample-and-hold period for SOC0 starts.
5. At the end of the sample-and-hold period for SOC0, the conversion for SOC0 begins. ADCxEXTMUX remains unchanged until the start of the next SOC sample-and-hold period.
6. At the end of the conversion for SOC0, SOC2 is the next pending SOC. SOC2's sample-and-hold period begins, and ADCSOC2CTL.EXTCHSEL is pushed onto the ADCxEXTMUX pins.
7. The SOC2 conversion begins, and there are no pending SOCs left. ADCxEXTMUX remains unchanged until a new SOC trigger arrives.

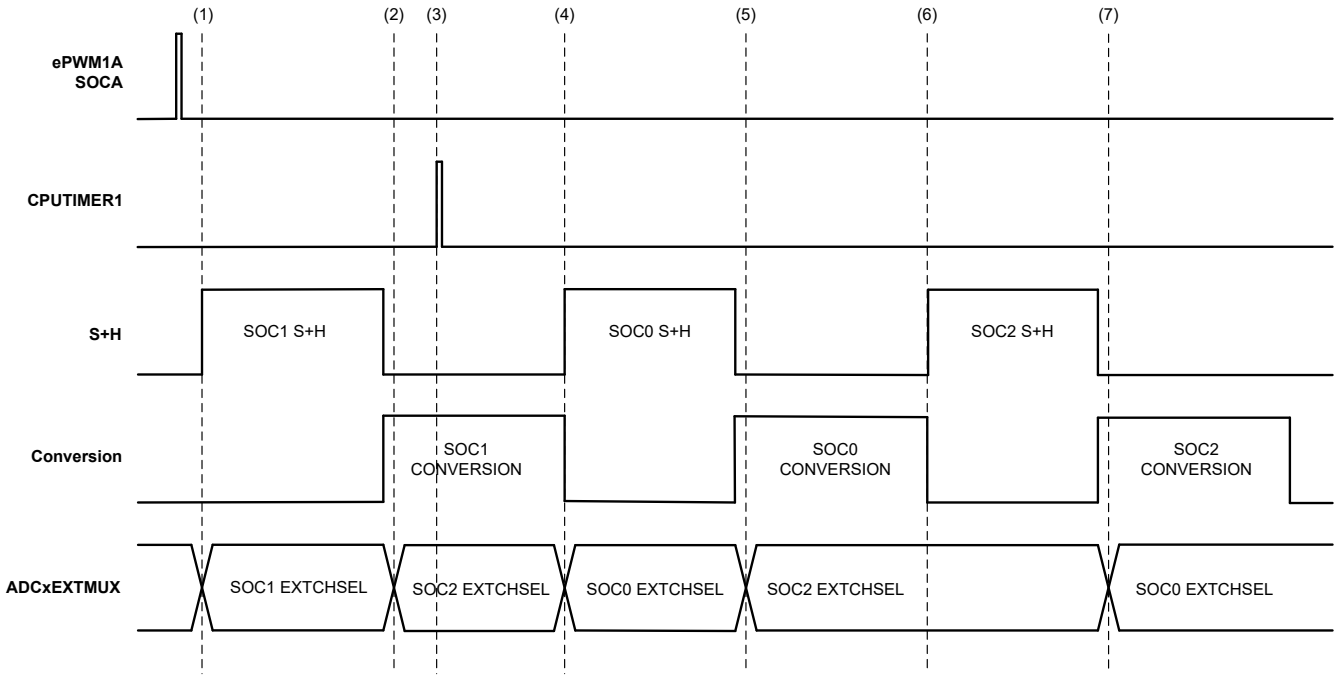


Figure 24-18. ADC External Channel Timing Example in Preselect Mode with Asynchronous Trigger

In Figure 24-18, the ADC is configured as follows:

- SOC1 and SOC2 are triggered from ePWM1A;
- SOC0 is triggered from CPUTIMER1, and has a high priority.

With this configuration, the ADC performs the SOC operation sequence in the following order:

1. The initial ePWM1A trigger arrives, setting the SOC1 and SOC2 flags to pending. SOC1 gains priority, and the SOC1 sample-and-hold period begins. The ADC pushes the value of ADCSOC1CTL.EXTCHSEL onto the ADCxEXTMUX pins at the same time when the SOC1 sample-and-hold begins.
2. At the end of the sample-and-hold for SOC1, the highest priority SOC that is pending is SOC2, so the ADC pushes the value of ADCSOC2CTL.EXTCHSEL onto the ADCxEXTMUX pins.
3. CPUTIMER1 issues a trigger asynchronously, setting the SOC0 flag to pending.
4. Since SOC0 has high priority, SOC0 converts next instead of SOC2. The ADC overwrites the previous speculative external mux selection (ADCSOC2CTL.EXTCHSEL) with ADCSOC0.EXTCHSEL when the sample-and-hold period for SOC0 starts. In situations like this where asynchronous triggers are possible, make sure to set the acquisition window size of the priority SOC large enough to allow for both external mux settling and internal channel settling.
5. At the end of the sample-and-hold period for SOC0, the highest priority SOC that is pending is SOC2, so the ADC pushes EXTCHSEL value for SOC2 onto the ADCxEXTMUX pins. SOC0 begins converting.
6. At the end of the SOC0 conversion, SOC2 is the next highest-priority pending SOC, and so SOC2's sample-and-hold period begins. The ADC again pushes ADCSOC2CTL.EXTCHSEL onto the ADCxEXTMUX pins, but since this is already the current value, the mux pins are unchanged.
7. At the end of the sample-and-hold period for SOC2, there are no SOCs pending. SOC0 has the next highest priority, since the ADC has been configured to give SOC0 high priority. The ADC pushes ADCSOC0CTL.EXTCHSEL onto the ADCxEXTMUX pins.

24.4 SOC Configuration Examples

The following sections provide some specific examples of how to configure the SOCs to produce some conversions.

24.4.1 Single Conversion from ePWM Trigger

To configure ADCA to perform a single conversion on channel ADCIN1 when the ePWM timer reaches the period match, a few things are necessary. First, ePWM3 must be configured to generate an SOCA or SOCB signal (in this statement, SOC refers to a signal in the ePWM module). See the *Enhanced Pulse Width Modulator Module (ePWM)* chapter on how to do this. Assume that SOCB was chosen.

SOC5 is chosen arbitrarily. Any of the 32 SOCs can be used.

Assuming a 100ns sample window is desired with a SYSCLK frequency of 200MHz, then the acquisition window duration must be 100ns/5ns = 20 SYSCLK cycles. The ACQPS field must be set to 20 - 1 = 19.

```
AdcaRegs.ADCSOC5CTL.bit.CHSEL = 1;      //SOC5 converts ADCINA1
AdcaRegs.ADCSOC5CTL.bit.ACQPS = 19;     //SOC5 uses a sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC5CTL.bit.TRIGSEL = 10;   //SOC5 begins conversion on ePWM3 SOCB
```

As configured, when ePWM3 matches the period and generates the SOCB signal, the ADC begins sampling channel ADCINA1 (SOC5) immediately if the ADC is idle. If the ADC is busy, ADCINA1 begins sampling when SOC5 gains priority (see Section 24.5). The ADC control logic samples ADCINA1 with the specified acquisition window width of 100ns. Immediately after the acquisition is complete, the ADC begins converting the sampled voltage to a digital value. When the ADC conversion is complete, the results are available in the ADCRESULT5 register (see Section 24.13 for exact sample, conversion, and result latch timings).

24.4.2 Oversampled Conversion from ePWM Trigger

To configure the ADC to oversample ADCINA1 4 times, we use the same configurations as the previous example, but apply them to SOC5, SOC6, SOC7, and SOC8.

```
AdcaRegs.ADCSOC5CTL.bit.CHSEL = 1;        //SOC5 converts ADCINA1
AdcaRegs.ADCSOC5CTL.bit.ACQPS = 19;       //SOC5 uses a sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC5CTL.bit.TRIGSEL = 10;     //SOC5 begins conversion on ePWM3 SOCB
AdcaRegs.ADCSOC6CTL.bit.CHSEL = 1;        //SOC6 converts ADCINA1
AdcaRegs.ADCSOC6CTL.bit.ACQPS = 19;       //SOC6 uses a sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC6CTL.bit.TRIGSEL = 10;     //SOC6 begins conversion on ePWM3 SOCB
AdcaRegs.ADCSOC7CTL.bit.CHSEL = 1;        //SOC7 converts ADCINA1
AdcaRegs.ADCSOC7CTL.bit.ACQPS = 19;       //SOC7 uses a sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC7CTL.bit.TRIGSEL = 10;     //SOC7 begins conversion on ePWM3 SOCB
AdcaRegs.ADCSOC8CTL.bit.CHSEL = 1;        //SOC8 converts ADCINA1
AdcaRegs.ADCSOC8CTL.bit.ACQPS = 19;       //SOC8 uses a sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC8CTL.bit.TRIGSEL = 10;     //SOC8 begins conversion on ePWM3 SOCB
```

As configured, when ePWM3 matches the period and generates the SOCB signal, the ADC begins sampling channel ADCINA1 (SOC5) immediately if the ADC is idle. If the ADC is busy, ADCINA1 begins sampling when SOC5 gains priority (see Section 24.5). Once the conversion is complete for SOC5, SOC6 begins converting ADCINA1 and the results for SOC5 are placed in the ADCRESULT5 register. All four conversions eventually are completed sequentially, with the results in ADCRESULT5, ADCRESULT6, ADCRESULT7, and ADCRESULT8 for SOC5, SOC6, SOC7, and SOC8, respectively.

Note

It is possible, but unlikely, that the ADC can begin converting SOC6, SOC7, or SOC8 before SOC5 depending on the position of the round-robin pointer when the ePWM trigger is received. See Section 24.5 to understand how the next SOC to be converted is chosen.

24.4.3 Multiple Conversions from CPU Timer Trigger

This example shows how to sample multiple signals with different acquisition window requirements. CPU1 Timer 2 is used to generate the trigger. To see how to configure the CPU timer, see the *System Control and Interrupts* chapter.

A good first step when designing a sampling scheme with many signals is to list out the signals and the required acquisition window. From this, calculate the necessary number of SYSCLK cycles for each signal, then the ACQPS register setting. This is shown in Table 24-8, where a SYCLK of 200MHz is assumed (5ns cycle time).

Table 24-8. Example Requirements for Multiple Signal Sampling

Signal Name	Acquisition Window Requirement	Acquisition Window (SYSCLK Cycles)	ACQPS Register Value
Signal 1	>120ns	120ns/5ns = 24	24 – 1 = 23
Signal 2	>444ns	444ns/5ns = 89 (round up)	89 – 1 = 88
Signal 3	>110ns	110ns/5ns = 22	22 – 1 = 21
Signal 4	>291ns	291ns/5ns = 59 (round up)	59 – 1 = 58

Next decide which ADC pins to connect to each signal. This is highly dependent on the application board layout. Once the pins are selected, determining the value of CHSEL is straightforward (see Table 24-9).

Table 24-9. Example Connections for Multiple Signal Sampling

Signal Name	ADC Pin	CHSEL Register Value
Signal 1	ADCINA5	5
Signal 2	ADCINA0	0
Signal 3	ADCINA3	3
Signal 4	ADCINA2	2

With the information tabulated, generate the SOC configurations:

```
AdcaRegs.ADCSOC0CTL.bit.CHSEL = 5;        //SOC0 converts ADCINA5
AdcaRegs.ADCSOC0CTL.bit.ACQPS = 23;       //SOC0 uses a sample duration of 24 SYSCLK cycles
AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 3;      //SOC0 begins conversion on CPU1 Timer 2
AdcaRegs.ADCSOC1CTL.bit.CHSEL = 0;        //SOC1 converts ADCINA0
AdcaRegs.ADCSOC1CTL.bit.ACQPS = 88;       //SOC1 uses a sample duration of 89 SYSCLK cycles
AdcaRegs.ADCSOC1CTL.bit.TRIGSEL = 3;      //SOC1 begins conversion on CPU1 Timer 2
AdcaRegs.ADCSOC2CTL.bit.CHSEL = 3;        //SOC2 converts ADCINA3
AdcaRegs.ADCSOC2CTL.bit.ACQPS = 21;       //SOC2 uses a sample duration of 22 SYSCLK cycles
AdcaRegs.ADCSOC2CTL.bit.TRIGSEL = 3;      //SOC2 begins conversion on CPU1 Timer 2
AdcaRegs.ADCSOC3CTL.bit.CHSEL = 2;        //SOC3 converts ADCINA2
AdcaRegs.ADCSOC3CTL.bit.ACQPS = 58;       //SOC3 uses a sample duration of 59 SYSCLK cycles
AdcaRegs.ADCSOC3CTL.bit.TRIGSEL = 3;      //SOC3 begins conversion on CPU1 Timer 2
```

As configured, when CPU1 Timer 2 generates an event, SOC0, SOC1, SOC2, and SOC3 eventually is sampled and converted, in that order. The conversion results for ACINA5 (Signal 1) are in ADCRESULT0. Similarly, The results for ADCINA0 (Signal 2), ADCINA3 (Signal 3), and ADCINA2 (Signal 4) are in ADCRESULT1, ADCRESULT2, and ADCRESULT3, respectively.

Note

There is a possibility, but unlikely, that the ADC can begin converting SOC1, SOC2, or SOC3 before SOC0 depending on the position of the round-robin pointer when the CPU Timer trigger is received. See Section 24.5 to understand how the next SOC to be converted is chosen.

24.4.4 Software Triggering of SOCs

At any point, whether or not the SOCs have been configured to accept a specific trigger, a software trigger can set the SOCs to be converted. This is accomplished by writing bits in the ADCSOCFRC1 register.

Software triggering of the previous example without waiting for the CPU1 Timer 2 to generate the trigger can be accomplished by the statement:

```
AdcaRegs.ADCSOCFRC1.all = 0x000F;        //set SOC flags for SOC0 to SOC3
```

24.5 ADC Conversion Priority

When multiple SOC flags are set at the same time, one of two forms of priority determines the converted order. The default priority method is round-robin. In this scheme, no SOC has an inherent higher priority than another. Priority depends on the round-robin pointer (RRPOINTER). The RRPOINTER reflected in the ADCSOCPRIORITYCTL register points to the last SOC converted. The highest priority SOC is given to the next value greater than the RRPOINTER value, wrapping around back to SOC0 after SOC15. At reset the value is 16 since 0 indicates a conversion has already occurred. When RRPOINTER equals 16 the highest priority is given to SOC0. The RRPOINTER is reset when the ADC module is reset or when the reset value is written to the SOCPRICTL register. The ADC module is reset by writing and clearing the SOFTPRES bit corresponding to the ADC instance.

An example of the round-robin priority method is given in Figure 24-19.

The SOCPRIORITY field in the ADCSOCPRIORITYCTL register can be used to assign high priority from a single to all of the SOCs. When configured as high priority, an SOC interrupts the round-robin wheel after any current conversion completes and inserts in as the next conversion. After the conversion completes, the round-robin wheel continues where the conversion was interrupted. If two high priority SOCs are triggered at the same time, the SOC with the lower number takes precedence.

High priority mode is assigned first to SOC0, then in increasing numerical order. The value written in the SOCPRIORITY field defines the first SOC that is not high priority. In other words, if a value of 4 is written into SOCPRIORITY, then SOC0, SOC1, SOC2, and SOC3 are defined as high priority, with SOC0 the highest.

An example using high priority SOC's is given in Figure 24-20.

A After reset, SOC0 is highest priority SOC; SOC7 receives trigger; SOC7 configured channel is converted immediately.

B RRPOINTER changes to point to SOC 7; SOC8 is now highest priority SOC.

C SOC2 & SOC12 triggers rcvd. simultaneously; SOC12 is first on round robin wheel; SOC12 configured channel is converted while SOC2 stays pending.

D RRPOINTER changes to point to SOC 12; SOC2 configured channel is now converted.

E RRPOINTER changes to point to SOC 2; SOC3 is now highest priority SOC.

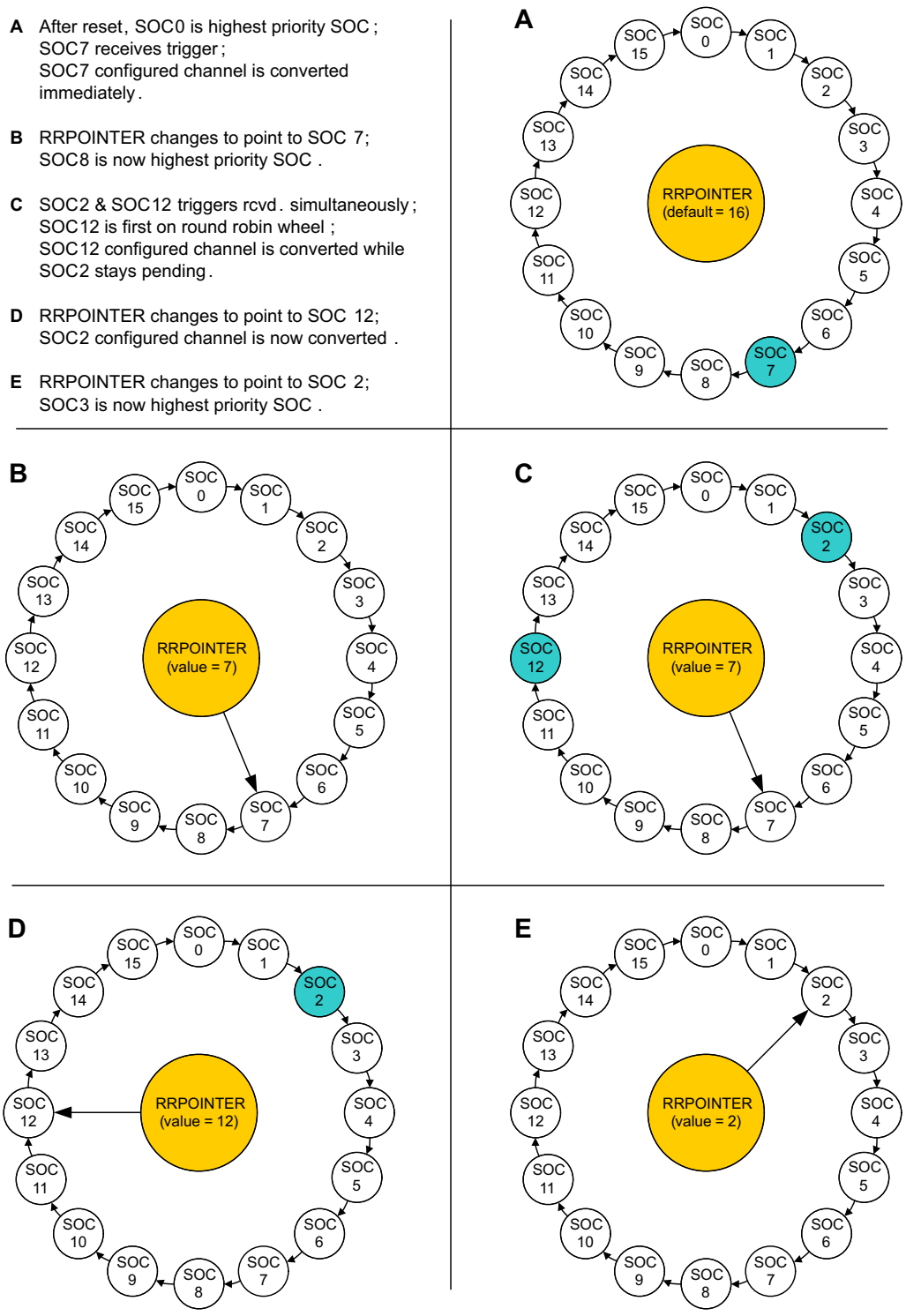


Figure 24-19. Round Robin Priority Example

Example when SOCPRIORITY = 4

A After reset, SOC4 is 1st on round robin wheel ;
SOC7 receives trigger ;
SOC7 configured channel is converted immediately .

B RRPOINTER changes to point to SOC 7;
SOC8 is now 1st on round robin wheel .

C SOC2 & SOC12 triggers rcvd . simultaneously ;
SOC2 interrupts round robin wheel and SOC 2 configured channel is converted while SOC 12 stays pending .

D RRPOINTER stays pointing to 7;
SOC12 configured channel is now converted .

E RRPOINTER changes to point to SOC 12;
SOC13 is now 1st on round robin wheel .

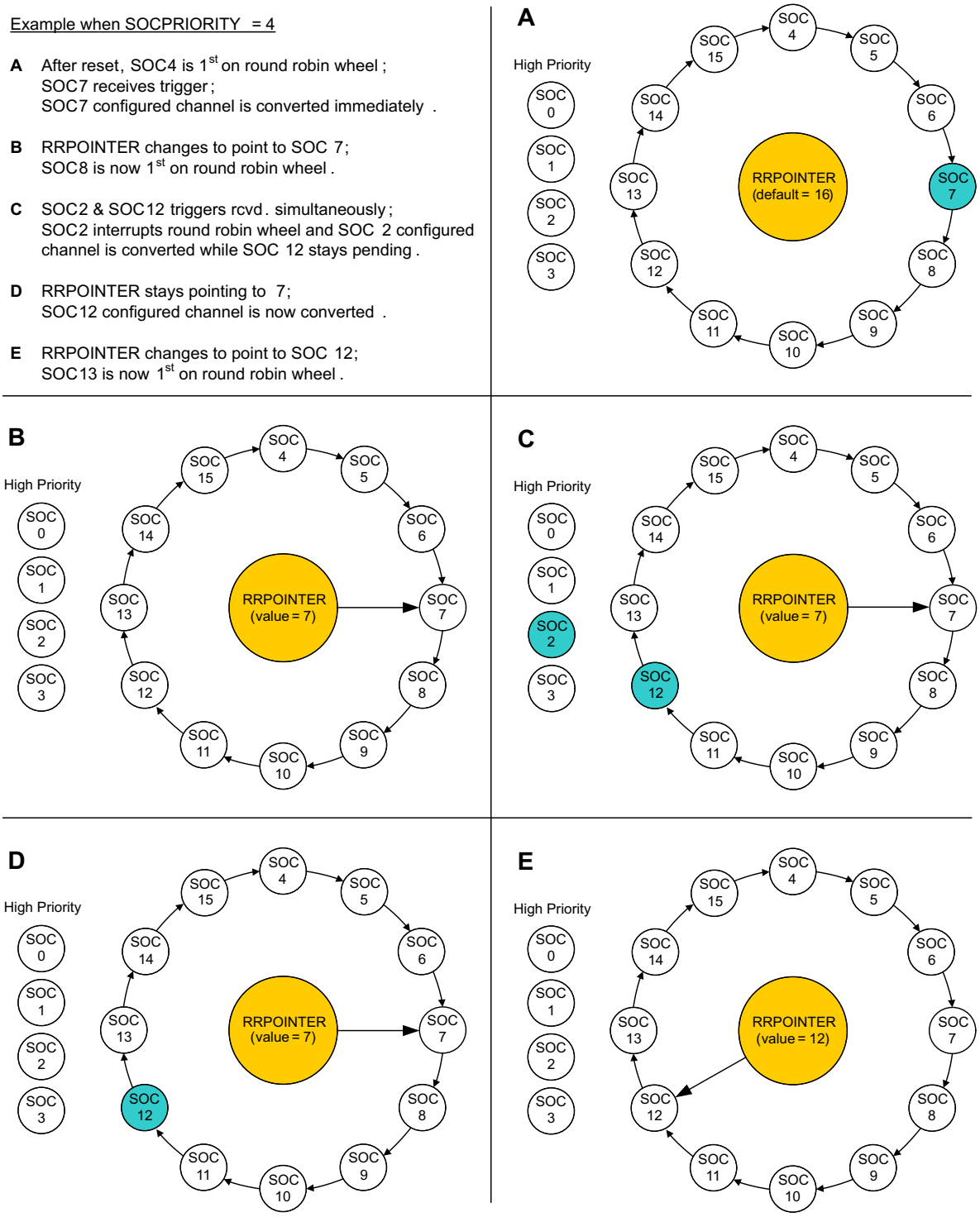


Figure 24-20. High Priority Example

24.6 Burst Mode

Burst mode allows a single trigger to walk through the round-robin SOCs one or more at a time. Setting the bit BURSTEN in the ADCBURSTCTL register configures the ADC wrapper for burst mode. This causes the TRIGSEL field to be ignored, but only for SOCs that are configured for round-robin operation (not high priority). Instead of the TRIGSEL field, all round-robin SOCs are triggered based on the BURSTTRIG field in the ADCBURSTCTL register. Upon reception of the burst trigger, the ADC wrapper does not set all round-robin SOCs to be converted, but only (ADCBURSTCTL.BURSTSIZE + 1) SOCs. The first SOC to be set is the SOC with the highest priority based on the round-robin pointer, and subsequent SOCs are set until BURSTSIZE SOCs have been set.

Note

When configuring the ADC for burst mode, the user is responsible for ensuring that each burst of conversions is allowed to complete before the next burst trigger is received. The value of (ADCBURSTCTL.BURSTSIZE + 1) must be less than or equal to the number of SOCs configured for round-robin priority. If the previous burst is not complete at the time when a new burst trigger arrives, for each SOC that was already pending and receives a new trigger, the corresponding overflow flag in ADCSOCOVF1 is set.

For example, if SOCPRIORITY = 12, that is, SOC12, SOC13, SOC14, and SOC15 are in round-robin, ADCBURSTCTL.BURSTSIZE setting must be ≤3 for burst mode to operate correctly.

24.6.1 Burst Mode Example

Burst mode can be used to sample a different set of signals on every other trigger. In the following example, ADCIN7 and ADCIN5 are converted on the first trigger from CPU1 Timer 2 and every other trigger thereafter. ADCIN2 and ACIN3 are converted on the second trigger from CPU1 Timer 2 and every other trigger thereafter. All signals are converted with 20 SYSCLK cycle wide acquisition windows, but different durations can be configured for each SOC as desired.

```
AdcaRegs.BURSTCTL.BURSTEN = 1;              //Enable ADC burst mode
AdcaRegs.BURSTCTL.BURSTTRIG = 3;            //CPU1 Timer 2 triggers burst of conversions
AdcaRegs.BURSTCTL.BURSTSIZE = 1;            //conversion bursts are 1 + 1 = 2 conversions long
AdcaRegs.SOCPRICTL.bit.SOCPRIORITY = 12;    //SOC0 to SOC11 are high priority
AdcaRegs.ADCSOC12CTL.bit.CHSEL = 7;         //SOC12 converts ADCINA7
AdcaRegs.ADCSOC12CTL.bit.ACQPS = 19;        //SOC12 uses sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC13CTL.bit.CHSEL = 5;         //SOC13 converts ADCINA5
AdcaRegs.ADCSOC13CTL.bit.ACQPS = 19;        //SOC13 uses sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC14CTL.bit.CHSEL = 2;         //SOC14 converts ADCINA2
AdcaRegs.ADCSOC14CTL.bit.ACQPS = 19;        //SOC14 uses sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC15CTL.bit.CHSEL = 3;         //SOC15 converts ADCINA3
AdcaRegs.ADCSOC15CTL.bit.ACQPS = 19;        //SOC15 uses sample duration of 20 SYSCLK cycles
```

When the first CPU1 Timer 2 trigger is received, SOC12 and SOC13 are converted immediately if the ADC is idle. If the ADC is busy, SOC12 and SOC13 are converted once the SOCs gain priority. The results for SOC12 and SOC13 are in ADCRESULT12 and ADCRESULT13, respectively. After SOC13 completes, the round-robin pointer gives the highest priority to SOC14. Because of this, when the next CPU1 Timer 2 trigger is received, SOC14 and SOC15 is set as pending and eventually converted. The results for SOC14 and SOC15 are in ADCRESULT14 and ADCRESULT15, respectively. Subsequent triggers continue to toggle between converting SOC12 and SOC13, and converting SOC14 and SOC15.

While the above example toggles between two sets of conversions, three or more different sets of conversions can be achieved using a similar approach.

24.6.2 Burst Mode Priority Example

An example of priority resolution using burst mode and high-priority SOCs is presented in Figure 24-21.

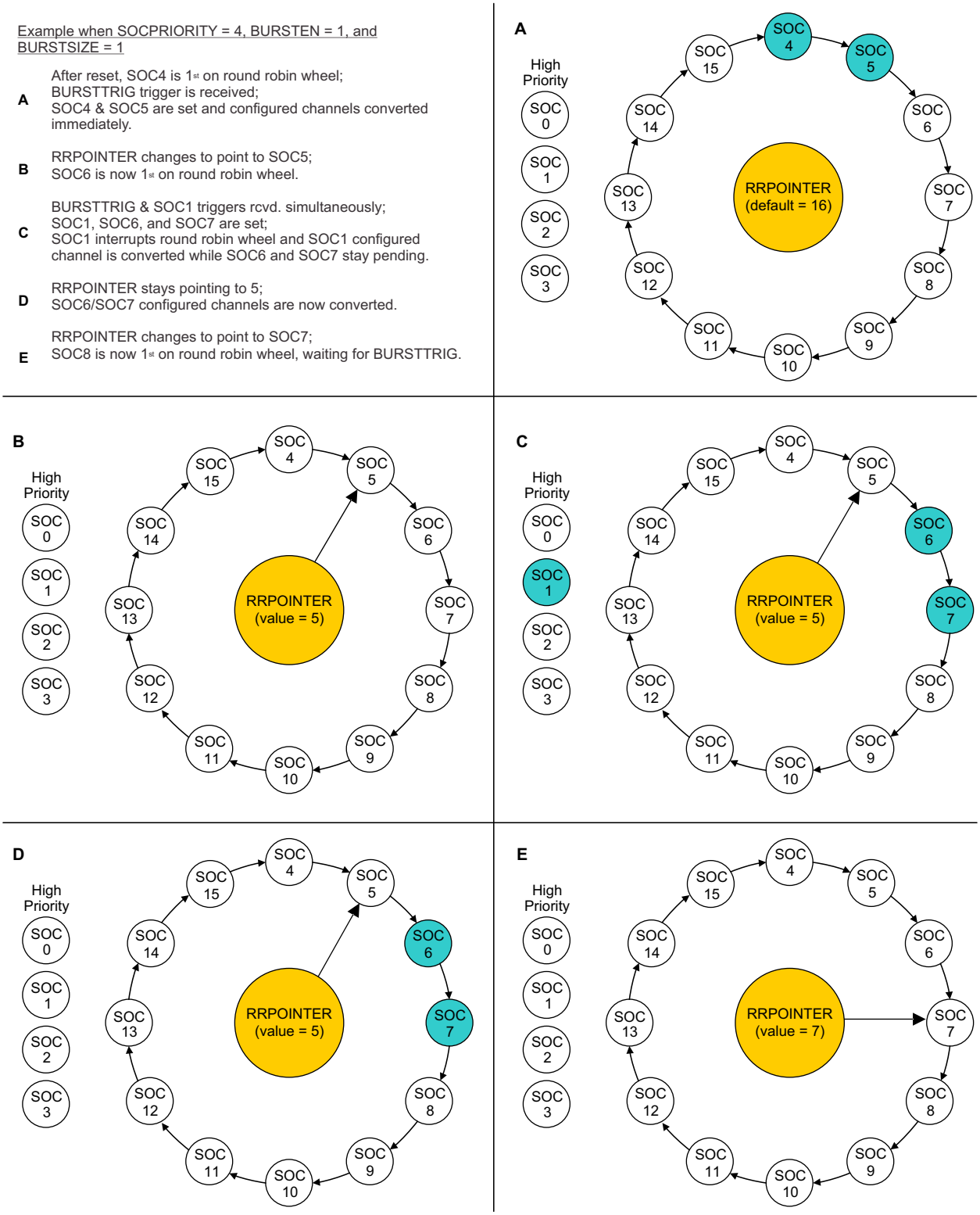


Figure 24-21. Burst Priority Example

24.7 EOC and Interrupt Operation

Each SOC has a corresponding end-of-conversion (EOC) signal. This EOC signal can be used to trigger an ADC interrupt. The ADC can be configured to generate the EOC pulse at either the end of the acquisition window or at the end of the voltage conversion. This is configured using the bit INTPULSEPOS in the ADCCTL1 register. See Section 24.13 for exact EOC pulse location.

Each ADC module has 4 configurable ADC interrupts. These interrupts can be triggered by any of the 32 EOC signals. The flag bit for each ADCINT can be read directly to determine if the associated SOC is complete or the interrupt can be passed on to the Interrupt Controller. Each ADCINT flag also has a corresponding ADCINTxRESULT flag. The ADCINTxRESULT flag is only set when results corresponding to the EOC are latched. This is useful for interrupt service routines or CLA tasks with early interrupt timing configured, allowing the application code to perform some pre-processing or setup work, and then acting on the ADC conversion result as soon as the result is latched.

It is also possible to generate an ADC interrupt based on a PPB oversampling logic event, such as when the sample count matches the configured limit. There are four oversampling interrupt (OSINT) flags available in each module for this purpose. Any of the ADCINT flags can be configured for an OSINT by configuring the INTxSEL field the corresponding ADCINTSELxNy register.

Note

The ADCCTL1.ADCBSY bit being clear does not indicate that all conversions in a set of SOCs have completed, only that the ADC is ready to process the next conversion. To determine if a sequence of SOCs is complete, link an ADCINT flag to the last SOC in the sequence and monitor that ADCINT flag.

Figure 24-22 shows a block diagram of the ADC interrupt structure.

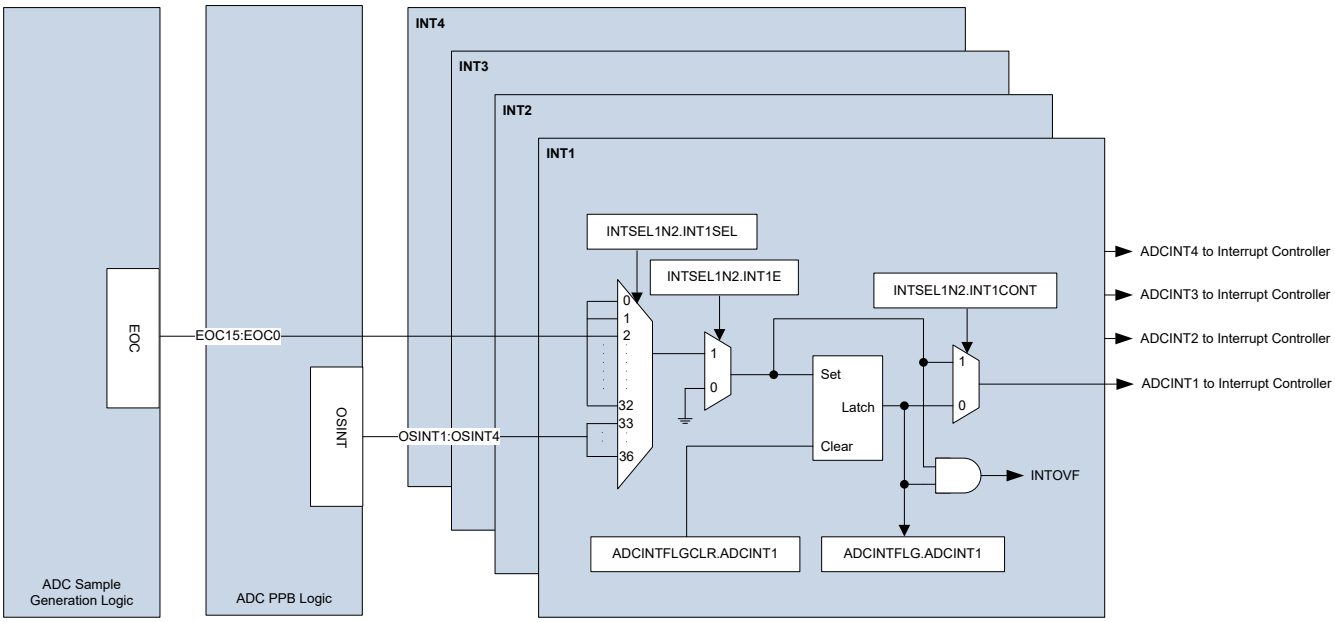


Figure 24-22. ADC EOC Interrupts

24.7.1 Interrupt Overflow

If the EOC signal sets a flag in the ADCINTFLG register, but that flag is already set, an interrupt overflow occurs. By default, overflow interrupts are not passed on to the Interrupt Controller module. When an overflow occurs on a given flag in the ADCINTFLG register, the corresponding flag in the ADCINOVF register is set. This overflow flag is only used to detect that an overflow has occurred; the flag does not block further interrupts from propagating to the Interrupt Controller module.

When an ADC interrupt overflow occurs, the application must check the appropriate ADCINTOVF flag inside the ISR or in the background loop and take appropriate action when an overflow is detected. The following code snippets demonstrate how to check the ADCINTOVF flag inside the ISR after attempting to clear the ADCINT flag.

```
// Clear the interrupt flag
AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;      //clear INT1 flag for ADC-A

// Check if an overflow has occurred
if(1 == AdcaRegs.ADCINTOVF.bit.ADCINT1)     //ADCINT overflow occurred
{
    AdcaRegs.ADCINTOVFCLR.bit.ADCINT1 = 1   //Clear overflow flag
    AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1   //Re-clear ADCINT flag
}
```

```
//
// Clear the interrupt flag
//
ADC_clearInterruptStatus(ADCA_BASE, ADC_INT_NUMBER1);

//
// Check if an overflow has occurred
//
if(true == ADC_getInterruptOverflowStatus(ADCA_BASE, ADC_INT_NUMBER1))
{
    ADC_clearInterruptOverflowStatus(ADCA_BASE, ADC_INT_NUMBER1);
    ADC_clearInterruptStatus(ADCA_BASE, ADC_INT_NUMBER1);
}
```

24.7.2 Continue to Interrupt Mode

The INTxCONT bits in the ADCINTSEL1N2 and ADCINTSEL3N4 registers configure how interrupts are handled when an ADCINTFLG has not yet been cleared from a prior interrupt. This mode is disabled by default and additional overlapping interrupts are not issued to the Interrupt Controller. By activating this mode, ADC interrupts always reach the Interrupt Controller. If interrupts occur while ADCINTFLG is set, the ADCINTOVF register remains set regardless of the configuration of the INTxCONT bits.

24.7.3 Early Interrupt Configuration Mode

Enabling early interrupt mode can allow the application to enter the ADC interrupt service routine before the ADC results are ready. This allows the application to do any necessary pre-work so that the application can act on the ADC results immediately when the ADC results become available. If the timing of the early interrupt is too early, then the application needs to waste time until the updated ADC results become available. To prevent this situation, the time the ADC interrupt is entered in early interrupt mode is configurable by way of the DELAY field in the ADCINTCYCLE register.

- To use the configurable interrupt time, the ADC must be in early interrupt mode. To achieve this, clear the bit INTPULSEPOS to 0 in ADCCTL1.
- The DELAY value in the ADCINTCYCLE register sets the number of additional SYSCLK cycles after the falling edge of the SOC pulse before the ADCINT flag is set.
- If the value of DELAY goes beyond EOC, the ADC interrupt is generated along with EOC.
- Writing values to DELAY when INTPULSEPOS is set to 1 does not have any effect on the interrupt generation.

To determine exactly when the ADC result has been latched into the ADCRESULT register, poll the ADCINTxRESULT flag bit in the ADCINTFLG register.

24.8 Post-Processing Blocks

Each ADC module contains four post-processing blocks (PPB). These blocks can be associated with any of the 32 RESULT registers using the ADCPPBxCONFIG.CONFIG bit field. The post-processing blocks have the ability to:

- Remove an offset associated with the ADCIN channel
- Subtract out a reference value
- Compute the delta between the current conversion result and the previous conversion.
- Aggregate successive samples using sum , max, and min calculations
- Automatically calculate average of oversampled conversions without CPU overhead, when sample count is a power of 2
- Transform the conversion result into an absolute value
- Flag a zero-crossing point, with the option to trip a PWM and generate an interrupt
- Flag a high or low compare limit, with the option to trip a PWM and generate an interrupt
- Digitally filter high or low compare results, to help prevent unwanted threshold trips
- Record the delay between the associated SOC trigger and when sampling actually begins

Figure 24-23 presents the structure of each PPB. Subsequent sections explain the use of each submodule.

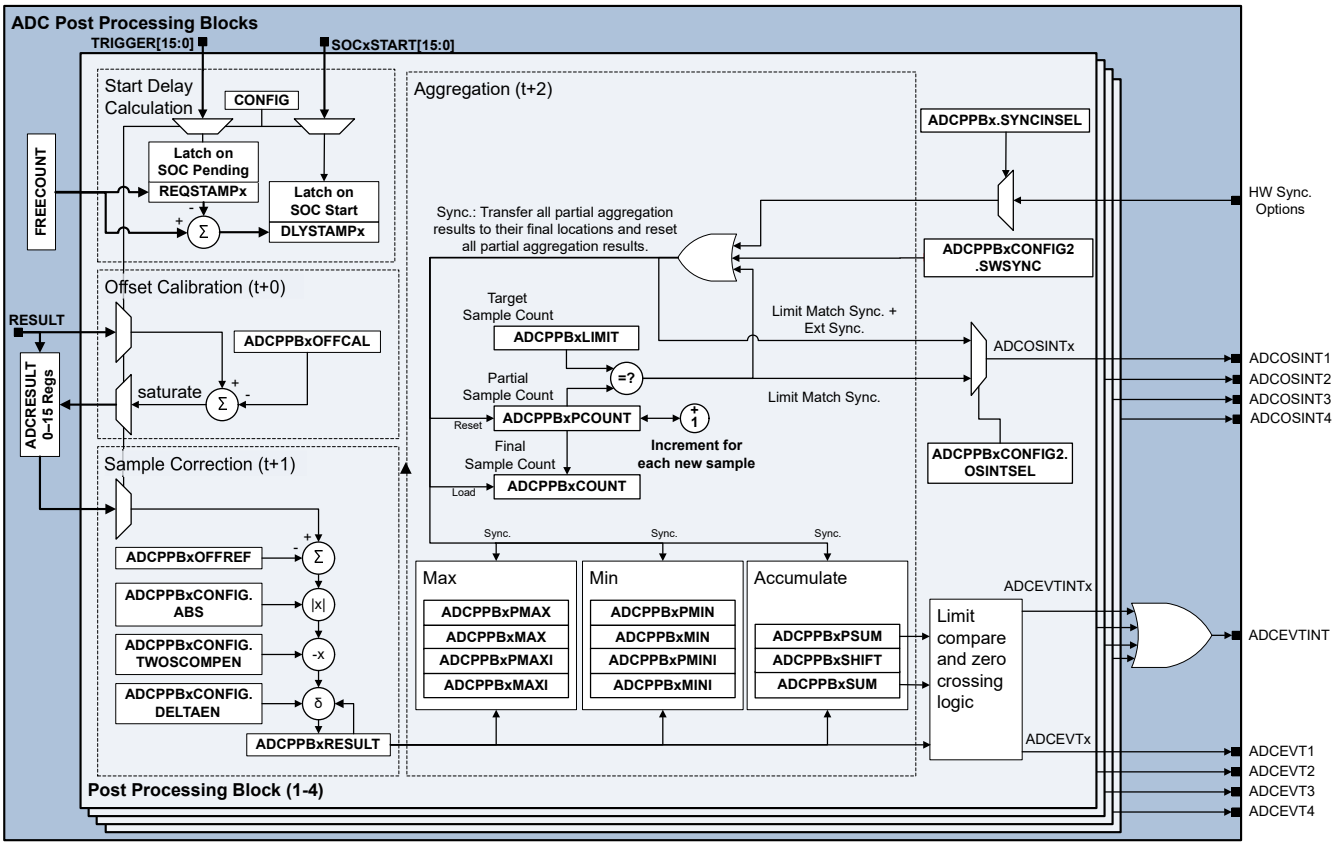


Figure 24-23. ADC PPB Block Diagram

24.8.1 PPB Offset Correction

In many applications, external sensors and signal sources produce an offset. A global trimming of the ADC offset is not enough to compensate for these offsets, which vary from channel to channel. The post-processing block can remove these offsets with zero overhead, saving numerous cycles in tight control loops.

Offset correction is accomplished by first pointing the ADCPPBxCONFIG.CONFIG to the desired SOC, then writing an offset correction value to the ADCPPBxOFFCAL.OFFCAL register. The post-processing block automatically adds or subtracts the value in the OFFCAL register from the raw conversion result and stores the value in the ADCRESULT register. This addition/subtraction saturates at 0 on the low end and either 4095 or 65535 on the high end for 12-bit or 16-bit mode, respectively.

Note

- Writing a 0 to the OFFCAL register effectively disables the offset correction feature, passing the raw result unchanged to the ADCRESULT register.
- To point multiple PPBs to the same SOC is possible. In this case, the OFFCAL value that is actually applied comes from the PPB with the lowest number.

24.8.2 PPB Error Calculation

In many applications, an error from a set point or expected value must be computed from the digital output of an ADC conversion. In other cases, a bipolar signal is necessary or convenient for control calculations. The PPB can perform these functions automatically, reducing the sample to output latency and reducing software overhead.

Error calculation is accomplished by first pointing the ADCPPBxCONFIG.CONFIG to the desired SOC, then writing a value to the ADCPPBxOFFCAL.OFFREF register. The post-processing block automatically subtracts the value in the OFFREF register from the ADCRESULT value and stores the value in the ADCPPBxRESULT register. This subtraction produces a sign-extended 32-bit result. It is also possible to selectively invert the calculated value before storing in the ADCPPBxRESULT register by setting the TWOSCOMPEN bit in the ADCPPBxCONFIG register.

If desired, the absolute value of the ADC result after the offset reference calculation can be obtained by setting the ADDCPPBxCONFIG.ABSEN bit. The absolute value is computed before evaluating the TWOSCOMPEN logic, so setting both TWOSCOMPEN and ABSEN always results in a negative value stored in ADCPPBxRESULT.

Note

- In 12-bit mode, do not write a value larger than 12 bits to the ADCPPBxOFFREF register.
- Since the ADCPPBxRESULT register is unique for each PPB, to point multiple PPBs to the same SOC and get different results for each PPB is possible.
- Writing a 0 to the ADCPPBxOFFREF register effectively disables the error calculation feature, passing the ADCRESULT value unchanged to the ADCPPBxRESULT register.

24.8.3 PPB Result Delta Calculation

The ADC's post-processing block has the capability to calculate the delta from the previous conversion sample. When enabled, the ADCPPBxRESULT register contains the difference between the current conversion result and the previous conversion. This delta is computed on the actual conversion result, not the result of the a prior delta calculation. The delta calculation occurs after OFFREF, TWOSCOMPEN and ABSEN calculations have been applied.

To enable result delta computation, write 1 to the DELTAEN bit in the ADCPPBxCONFIG register.

24.8.4 PPB Limit Detection and Zero-Crossing Detection

Many applications perform a limit check against the ADC conversion results. The PPB can automatically perform a check against high and low limits, or whenever ADCPPBxRESULT changes sign. Based on these comparisons, the PPB can generate a trip to the PWM and an interrupt automatically, lowering the sample to ePWM latency and reducing software overhead. This functionality also enables safety-conscious applications to trip the ePWM based on an out-of-range ADC conversion without any CPU intervention.

To enable this functionality, first point the ADCPPBxCONFIG.CONFIG to the desired SOC, then write a value to one or both of the registers ADCPPBxTRIPHI.LIMITHI and ADCPPBxTRIPLO.LIMITLO (zero-crossing detection does not require further configuration). Whenever these limits are exceeded, the PPBxTRIPHI bit or PPBxTRIPLO bit is set in the ADCEVTSTAT register. Note that the PPBxZERO bit in the ADCEVTSTAT register is gated by end-of-conversion (EOC), not by the sign change in the ADCPPBxRESULT register. The ADCEVTCLR register has corresponding bits to clear these event flags. The ADCEVTSEL register has corresponding bits which allow the events to propagate through to the PWM. The ADCEVTINTSEL register has corresponding bits that allow the events to propagate through to the Interrupt Controller.

One Interrupt Controller interrupt is shared between all the PPBs for a given ADC module as shown in Figure 24-24.

Figure 24-25 illustrates the ADC limit compare and zero-crossing logic.

Note

- If different actions need to be taken for different PPB events from the same ADC module, then the ADCEVTINT ISR has to read the PPB event flags in the ADCEVTSTAT register to determine which event caused the interrupt.
- If different ePWM trips need to be generated separately for high compare, low compare, and zero-crossing, this can be achieved by pointing multiple PPBs to the same SOC.
- The zero-crossing detect circuit considers a result of zero to be positive.

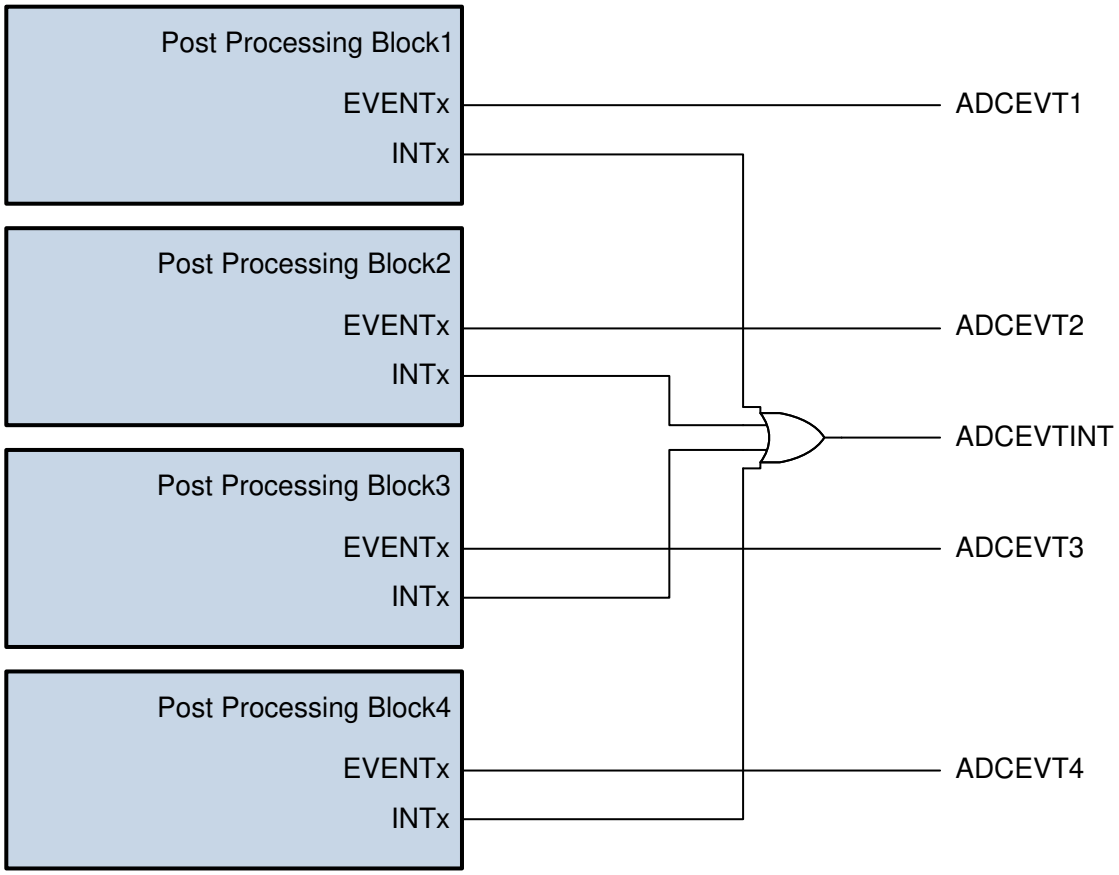


Figure 24-24. ADC PPB Interrupt Event

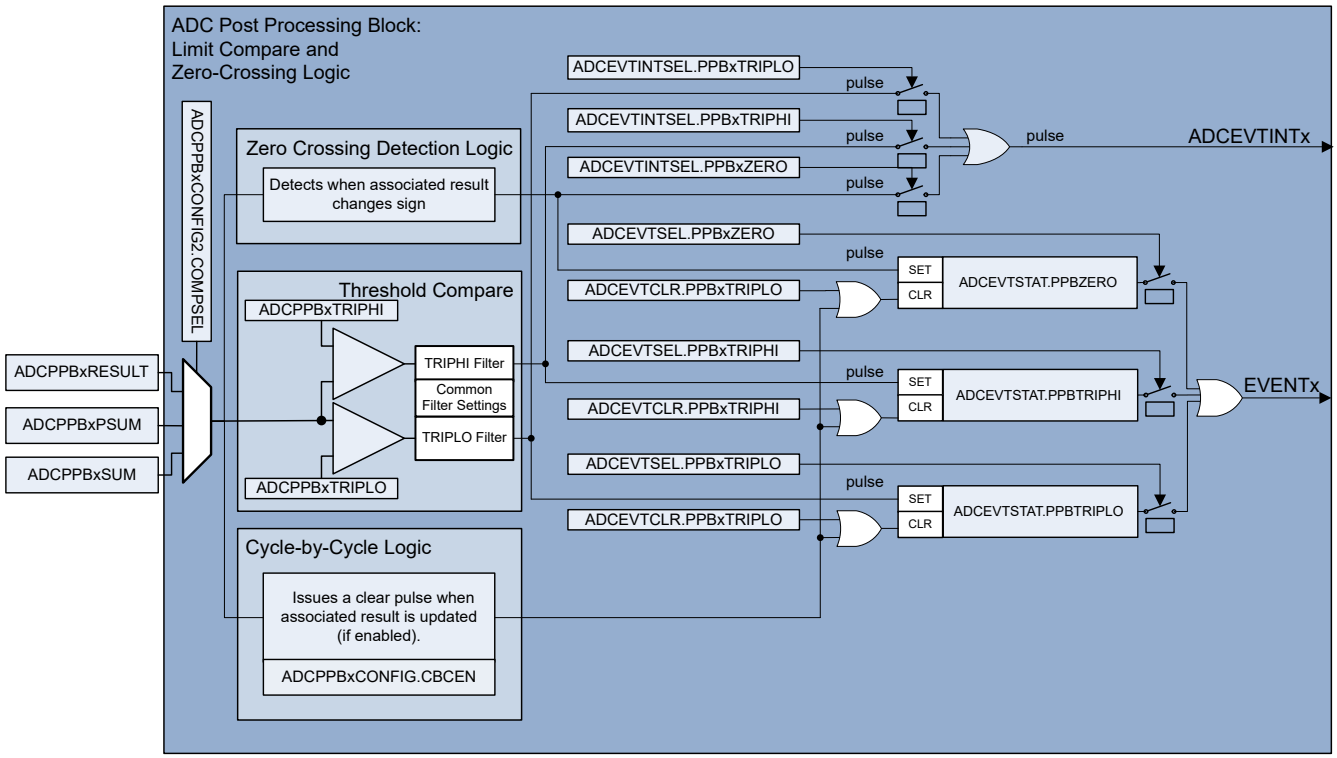


Figure 24-25. ADC PPB Limit Compare and Zero-Crossing Logic

24.8.4.1 PPB Digital Trip Filter

The ADC provides digital filters on the TRIPHI and TRIPLO signals to help prevent unwanted threshold trips. Each digital filter works on a window of FIFO samples (SAMPWIN) taken from the threshold comparator output. The filter output resolves to the majority value of the sample window, where majority is defined by the threshold (THRESH) value. If the majority threshold is not satisfied, the filter output remains unchanged. Figure 24-26 shows a block diagram of the PPB digital trip filter.

For proper operation, the value of THRESH must be greater than SAMPWIN / 2, and less than or equal to SAMPWIN.

A prescale function (CLKPRESCALE) determines the filter sampling rate. The filter FIFO captures one sample for every CLKPRESCALE cycles of SYSCLK. Old data from the FIFO is discarded.

The digital trip filters are disabled by default. To enable and configure the digital trip filters, write the desired values to the ADCPPBTRIPxFILCTL and ADCPPBTRIPxFILCLKCTL registers. Note that for SAMPWIN, THRESH and PRESCALE, the filter adds 1 to the value in the register field (for example, write 15 to SAMPWIN to enable a filter sample window of 16). To clear the FIFO and re-initialize the filter, write 1 to the FILINIT bit in the ADCPPBTRIPxFILCTL register.

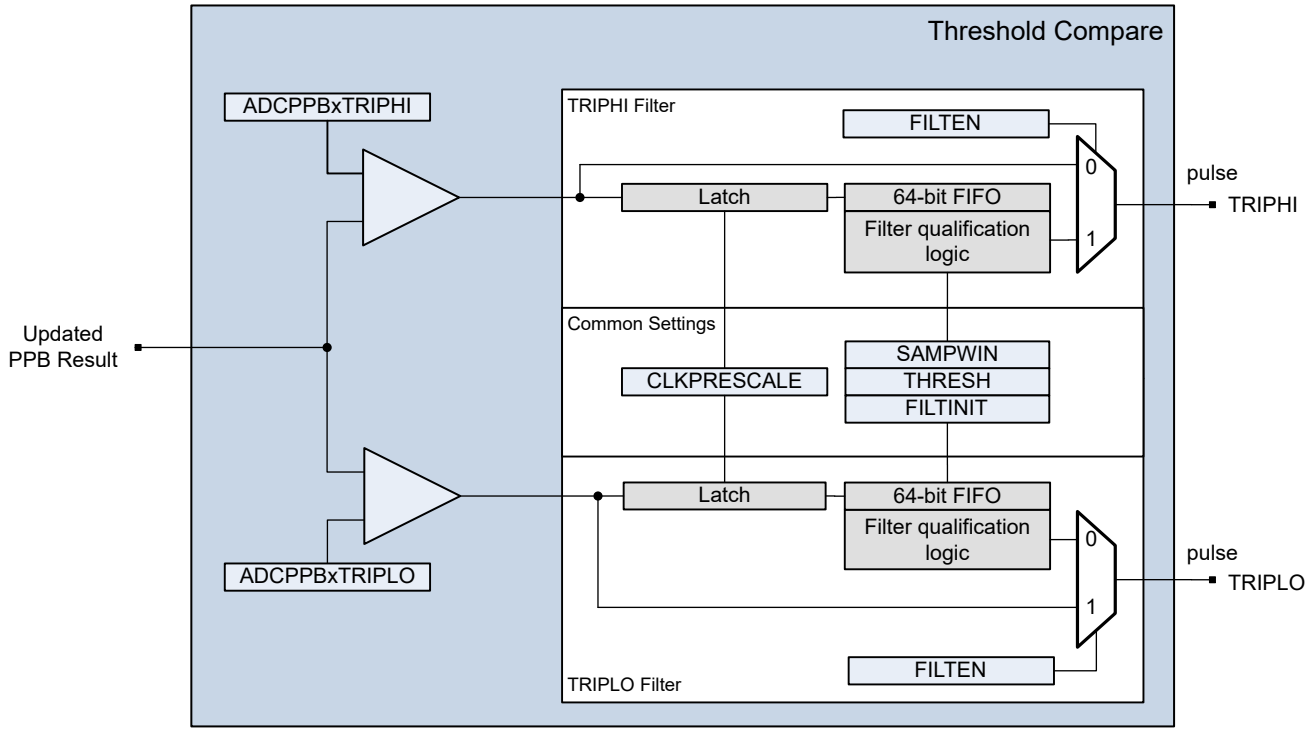


Figure 24-26. ADC PPB Limit Filter Logic

24.8.5 PPB Sample Delay Capture

When multiple control loops are running asynchronously on the same ADC, there is a chance that an ADC request from two or more loops collide, causing one of the samples to be delayed. This shows up as a measurement error in the system. By knowing when this delay occurs and the amount of delay that has occurred, software can employ extrapolation techniques to reduce the error.

To this effect, each PPB has the field DLYSTAMP in the ADCPPBxSTAMP register. This field contains the number of SYSCLK cycles between when the associate SOC was triggered and when the SOC began converting.

This is achieved by having a global 12-bit free running counter based off of SYSCLK, which is in the field FREECOUNT in the ADCCOUNTER register. When the trigger for the associated SOC arrives, the value of this counter is loaded into the bit field ADCPPBxTRIPLO.REQSTAMP. When the actual sample window for that SOC begins, the value in REQSTAMP is subtracted from the current FREECOUNT value and stored in DLYSTAMP.

Note

If more than 4096 SYSCLK cycles elapse between the SOC trigger and the actual start of the SOC acquisition, the FREECOUNT register can overflow more than once, leading to incorrect DLYSTAMP value. Be cautious when using very slow conversions to prevent this from happening.

The sample delay capture does not function, if the associated SOC is triggered using software. The sample delay capture, however, correctly records the delay, if the software triggering of a different SOC causes the SOC associated with the PPB to be delayed

24.8.6 PPB Oversampling

This ADC has built-in support for oversampling in the post-processing block, including an accumulator, min/max for peak detection, and outlier removal. The oversampling support module exists at the output of the sample correction module, as shown in Figure 24-23. The oversampling module works by accumulating results in partial registers until either the sample count limit defined in the ADCPPBxLIMIT register is reached, an external hardware sync event occurs, or the software forces a sync event by writing to the SWSYNC bit in the ADCPPBxCONFIG2 register.

24.8.6.1 Accumulation, Minimum, Maximum, and Average Functions

At the end of each ADC sample conversion, the PPB updates the partial result registers ADCPPBxPSUM, ADCPPBxPMIN, and ADCPPBxPMAX with the newly processed conversion result from the ADCPPBxRESULT register, and the partial conversion count register (ADCPPBxPCOUNT) increments by 1. When the partial conversion count equals the limit defined in ADCPPBxLIMIT, or the PPB receives a hardware or software sync signal, the PPB takes the following actions:

1. The PPB loads the values of the respective partial result registers into the final result registers ADCPPBxPSUM, ADCPPBxPMIN, and ADCPPBxPMAX.
2. The PPB loads the partial count in ADCPPBxPCOUNT into the final conversion count register ADCPPBxCOUNT.
3. The partial count register and partial result registers reset to zero.
4. The ADC generates an oversampling interrupt (OSINTx) event pulse, which triggers a CPU interrupt if so configured in the ADCINTSEL1N2 or ADCINTSEL3N4 registers.

The PPB can also be configured to generate an oversampling interrupt when there is a hardware or software sync event. To trigger an OSINTx pulse when a sync event occurs, write 1 to the OSINTSEL bit in the ADCPPBxCONFIG2 register.

The PPB can automatically compute the average of the accumulated samples if ADCPPBxLIMIT is set to a power of 2 (up to a maximum of 1024 samples). To perform automatic averaging over 2^n samples, set the SHIFT field in the ADCPPBxCONFIG2 register to n. When this field is set, the PPB divides the value of ADCPPBxPSUM by 2^n before loading into ADCPPBxSUM.

To compute an average from the accumulated sum when the number of samples is not a power of 2, divide the value of ADCPPBxSUM by the value of ADCPPBxCOUNT using the CPU.

Note

When using a sync signal to the repeater module and post-processing block to reset the ADC, note that the repeater sync signal does not stop or abort any pending SOCs. If both sync signals are issued simultaneously, any additional pending SOCs can propagate through the post-processing block after the sync signal has been issued. To fully clear or reset the ADC when using the repeater and PPB accumulation logic together:

1. Disable the repeater module trigger source.
2. Reset the trigger repeater by issuing a sync signal to the repeater module.
3. Wait for any pending SOCs to complete.
4. Finally, issue a sync signal to the post-processing block to complete the ADC reset.

24.8.6.2 Outlier Rejection

The post-processing block enables the application to easily perform outlier rejection, by eliminating the largest and smallest samples during each SOC burst. To eliminate outliers, the following formula can be used in a software routine or ISR:

$$\text{Average} = \frac{(\text{ADCPPBxSUM} - \text{ADCPPBxMAX} - \text{ADCPPBxMIN})}{(\text{ADCPPBxCOUNT} - 2)} \quad (14)$$

24.9 Result Safety Checker

For safety-critical applications, this device provides the ability to automatically compare ADC conversion results from multiple ADC modules against each other for consistency. The number of available safety checker tiles is specified in the device data sheet. Each ADC checker tile captures conversion results from the associated ADCs as soon as the conversions are complete, and compares the absolute value of the difference to the configured tolerance. If the computed delta is out of range, the checker can generate a trip event signal that is sent to an ePWM or output crossbar, and can also trigger an CPU interrupt. Figure 24-27 illustrates the structure and operation of an ADC result safety checker tile.

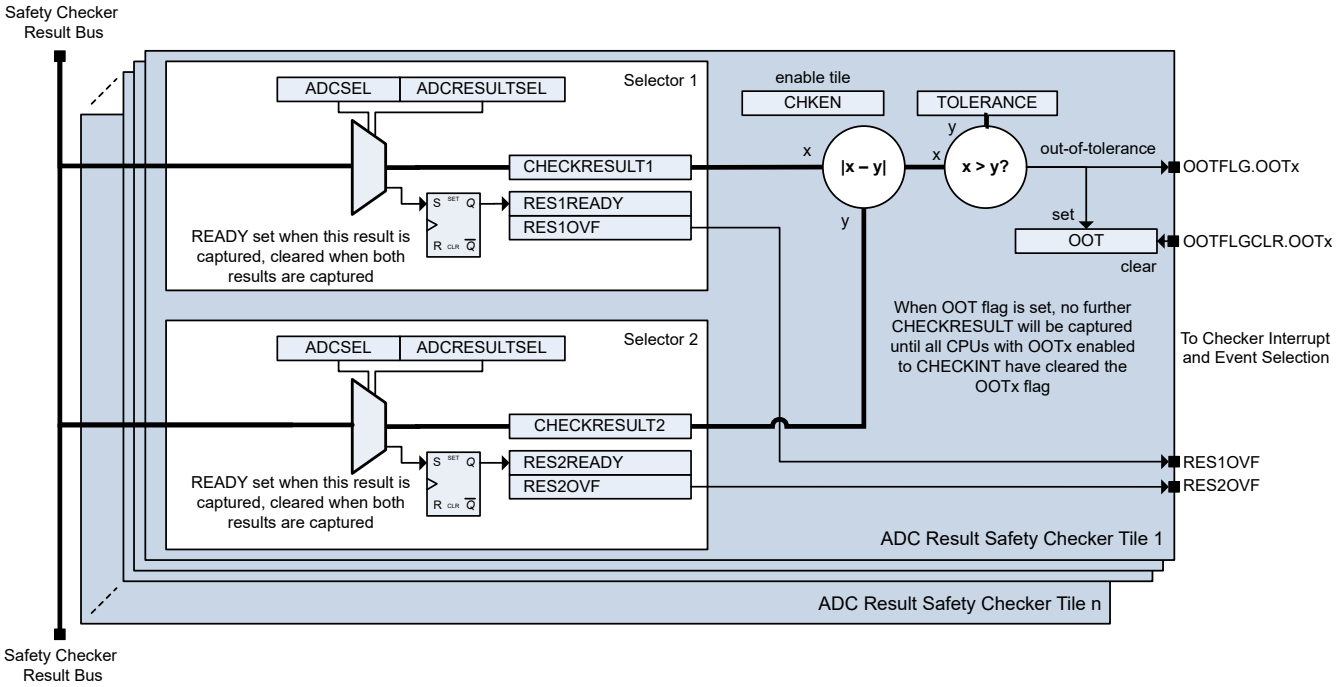


Figure 24-27. ADC Safety Checker Tile Diagram

24.9.1 Result Safety Checker Operation

Each ADC result safety checker tile can be configured to automatically compare two ADC conversion results against a set tolerance value, and generate an interrupt if an out-of-tolerance (OOT) event occurs. The selected results can be from the same ADC instance, or from different ADCs. The safety checker tiles exist outside of the ADC module, enabling the user application to compare results from two separate ADCs. To enable result safety checking, the application must first configure one or more ADCs to enable output of the desired results to the safety checker bus, and then configure the safety checker tile to compare those results.

To configure an ADC result safety checker tile:

1. Enable output of the desired ADC results for each ADC by writing to the ADC_REGSn.ADCSAFECHECKRESEN register. For each SOC, any one of the conversion result, PPB result, or PPB sum can be enabled for output to the safety checker bus by writing to ADCSAFECHECKRESEN.SOCxCHKEN.
2. Select the first result to compare by writing to the ADC_SAFECHECK_REGSn.ADCRESSEL1 register. Write to the ADCRESSEL1.ADCSEL field to select the ADC instance to test, and write to the ADCRESSEL1.ADCRESULTSEL field to select the corresponding SOC conversion result from that ADC to compare.
3. Select the second result to compare by writing to the ADC_SAFECHECK_REGSn.ADCRESSEL2 register. Write to ADCRESSEL2.ADCSEL to select the ADC instance to test, and write to ADCRESSEL2.ADCRESULTSEL to select the SOC conversion result from that ADC to compare. Any one of the SOC results, PPB module results or PPB sums can be selected in the ADCRESULTSEL field.
4. Configure the checker tolerance by writing to the ADC_SAFECHECK_REGSn.TOLERANCE register. The safety check result is out of tolerance if the difference between the two conversion results exceeds the value configured in TOLERANCE.
5. Enable the checker tile by writing 1 to ADC_SAFECHECK_REGSn.CHECKCONFIG.CHKEN.
6. Optionally, write 1 to ADC_SAFECHECK_REGSn.CHECKCONFIG.SWSYNC to force a reset of any currently set result safety checker event flags.

Each of the two result selectors waits for the configured ADC result to become available on the bus. When an ADC result becomes available, the checker reads the result into the CHECKRESULTx register, and sets the CHECKSTATUS.RESxRDY flag. Once both results are available, the checker clears the RESxRDY flags, compares the two results against each other, and sets the CHECKSTATUS.OOT flag if the configured tolerance is exceeded. This flag is also reflected in the corresponding bit for the selected checker tile in the OOTFLG register, and can be cleared by writing 1 to the corresponding bit in the OOTFLGCLR register. The checker does not perform any new comparisons while the OOT flag is set; to enable new comparisons, the flag must be cleared.

If two conversion results arrive in one selector before the other selector result becomes available, the checker sets the RESxOVF flag. This overflow flag does not prevent the comparison from occurring—the flag is for information only. To clear the overflow flag, write 1 to the corresponding bit for the selected checker tile in the RESxOVFCLR register.

24.9.2 Result Safety Checker Interrupts and Events

Each ADC result safety checker tile can generate an interrupt signal from out-of-tolerance (OOT) flags and result overflow flags (RESxOVF). These events are aggregated into a single interrupt signal, CHECKINT. Figure 24-28 shows a block diagram of the result checker interrupt aggregation. To enable a checker flag as a source for CHECKINT, write 1 to the corresponding bit in the CHECKINTSEL1, CHECKINTSEL2 or CHECKINTSEL3 register. To clear a previously set checker interrupt flag, write 1 to the CHECKINTFLGCLR.CHECKINTCLR register.

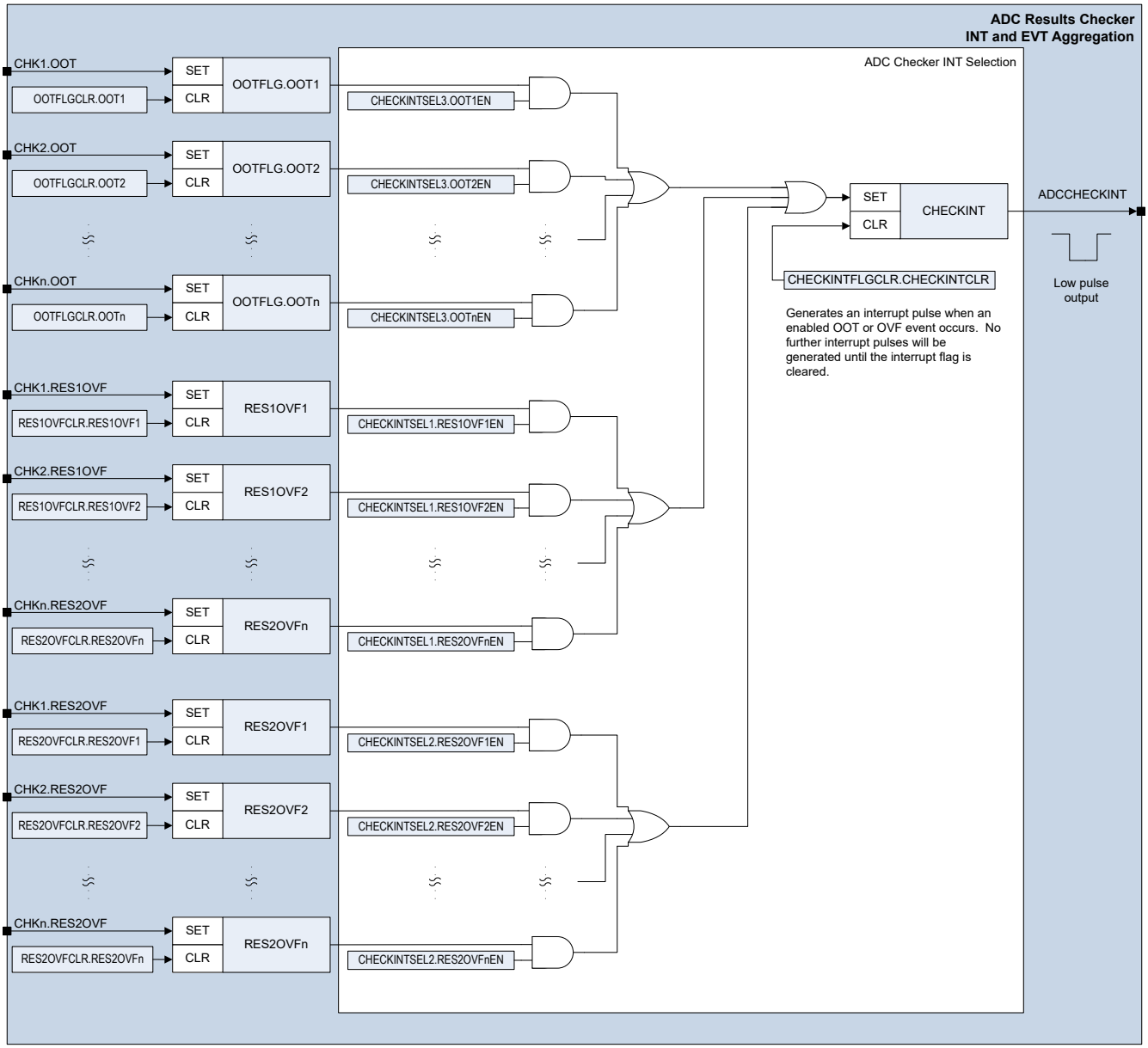


Figure 24-28. ADC Result Checker Interrupt Aggregation

In addition, safety checker tiles can also generate events that can be sent to the X-BAR, so that automatic hardware actions such as an ePWM trip can be generated. This device has 4 checker event signals (CHECKEVTx) that can be generated. For each event signal, any number of checker tile OOT or OVF flags can be aggregated into the single event signal. Figure 24-29 shows a block diagram of the ADC result checker event aggregation. To enable a checker flag as a source for a CHECKEVT signal, write 1 to the corresponding bit in the CHECKEVTxSEL1, CHECKEVTxSEL2 or CHECKEVTxSEL3 registers.

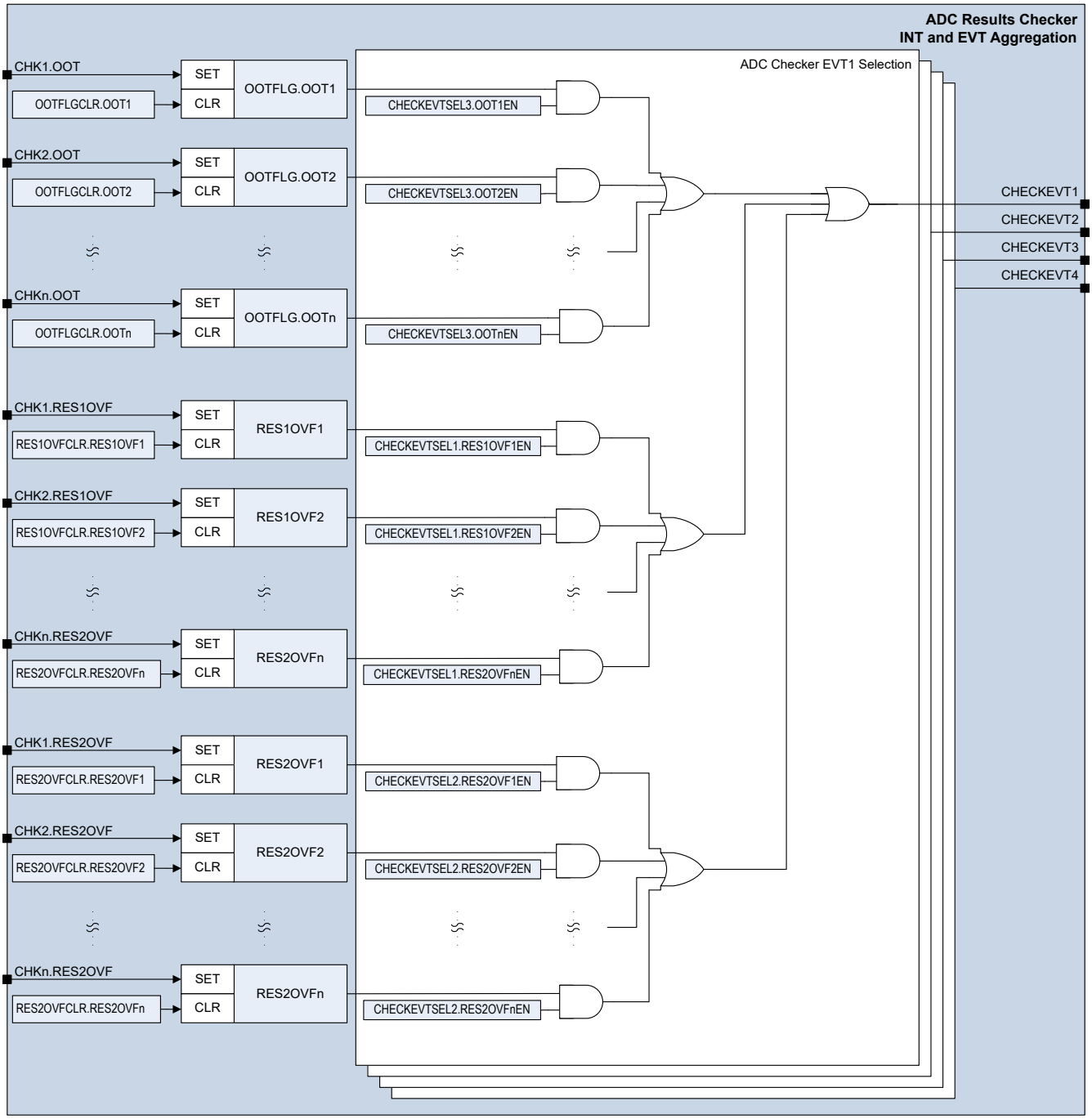


Figure 24-29. ADC Result Checker Event Aggregation

24.10 Opens/Shorts Detection Circuit (OSDETECT)

The opens/shorts detection circuit (OSDETECT) can be used to detect pin faults in the system. The circuit connects to the ADC input after the channel select multiplexer but before the S+H circuit as shown in Figure 24-30.

Note

- The divider resistance tolerances can vary widely; hence, this feature must not be used to check for conversion accuracy.
- See the data sheet for implementation and availability of analog input channels.
- Due to high drive impedance, a S+H duration much longer than the ADC minimum is needed.

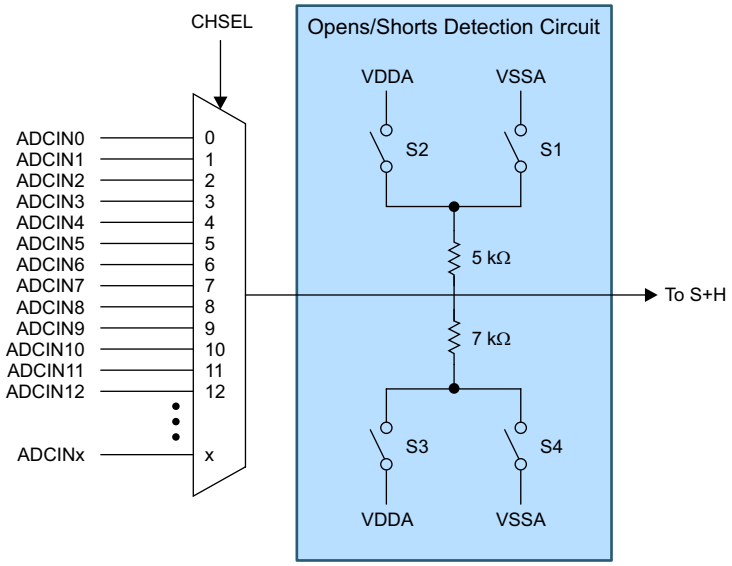


Figure 24-30. Opens/Shorts Detection Circuit

The circuit can be operated by writing a value to the DETECTCFG field in the ADCOSDETECT register. This causes the circuit to source a voltage onto the input during the S+H phase of any conversion. The voltage and drive strength of the OSDETECT circuit for different DETECTCFG settings is given in Table 24-10.

Table 24-10. DETECTCFG Settings

ADCOSDETECT. DETECTCFG	Source Voltage	S4	S3	S2	S1	Drive Impedance
0	Off	Open	Open	Open	Open	Open
1	Zero Scale	Closed	Open	Open	Closed	5K || 7K
2	Full Scale	Open	Closed	Closed	Open	5K || 7K
3	5/12 VDDA	Open	Closed	Open	Closed	5K || 7K
4	7/12 VDDA	Closed	Open	Closed	Open	5K || 7K
5	Zero Scale	Open	Open	Open	Closed	5K
6	Full Scale	Open	Open	Closed	Open	5K
7	Zero Scale	Closed	Open	Open	Open	7K

24.10.1 Implementation

A representative circuit with the OSDETECT implementation consists of the signal source with series resistance R_S, shunt capacitor C_P, the equivalent OSDETECT resistance $R_{OSDETECT}$ and voltage $V_{OSDETECT}$ is shown in Figure 24-31 and can be used as a basis to calculate the signal level going in to the sampling capacitor. $R_{OSDETECT}$ and $V_{OSDETECT}$ are the equivalent input resistance and voltage source contributed by the OSDETECT circuit with values shown in Table 24-10 for the different configuration settings. Refer to Figure 24-31 when deriving the input signal to S/H if signal source V_S is driving while the OSDETECT feature is enabled.

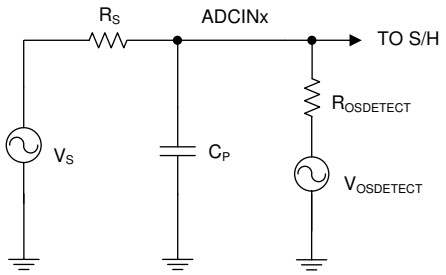


Figure 24-31. Input Circuit Equivalent with OSDETECT Enabled

The input impedance R_S and C_P are integral parts of the signal source or can have been implemented in the design to precondition the signal or to control signal settling time to meet S/H requirements. The input path has to be considered when using the OSDETECT feature, as this affects the conversion results. For instance, driving an input signal when this feature is enabled connects signal V_S to the OSDETECT circuit through R_S and affects the ADC results. Larger C_P values (in the order greater than hundreds of pF) require using higher ACQPS to make sure the signal at the input has settled prior to conversion.

To enable the circuit:

1. Configure the ADC for conversion (for example, channel, SOC, ACQPS, prescaler, trigger, and so on).
2. Set up the ADCOSDETECT register for the desired voltage divider connection as shown in Table 24-10.
3. Initiate a conversion and inspect the conversion result.

Interpret the results based on what is driving on the input side and what are the values of R_S and C_P. If the V_S signal can be disconnected from the input pin, the circuit can be used to detect open and shorted input pins as described in the following sections.

24.10.2 Detecting an Open Input Pin

By cycling through the various OSDETECT settings, the input signal is pulled towards the sourced voltages. An input with good drive strength (pin not open) is minimally affected. However, if the pin is open, the sampled voltages is close to the source voltages specified in Table 24-10.

24.10.3 Detecting a Shorted Input Pin

By cycling through the various OSDETECT settings, the input signal is pulled towards the sourced voltages. An input with finite drive strength (pin not shorted) is pulled toward each sourced voltage. However, if the pin is shorted, the signal remains at the same voltage.

24.11 Power-Up Sequence

Upon device power-up or system level reset, the ADC is powered down and disabled. When powering up the ADC, use the following sequence:

1. Set the bit to enable the desired ADC clock in the PCLKCR13 register.
2. Set the desired ADC clock divider in the PRESCALE field of ADCCTL2.
3. Power up the ADC by setting the ADCPWDNZ bit in ADCCTL1.
4. Allow a delay before sampling. See the data sheet for the necessary time.

If multiple ADCs are powered up simultaneously, steps 1 and step 3 can each be done for all ADCs in one write instruction. Also, only one delay is necessary as long as the delay occurs after all the ADCs have begun powering up.

24.12 ADC Calibration

During the fabrication and test process, Texas Instruments calibrates the gain, offset, and linearity of the ADCs and the offset of the buffered DACs. These trim settings are stored in M0 RAM during device boot, and can be loaded using C-callable functions.

- The Device_cal() function copies the trim values for ADC and DAC offset from M0 RAM to the respective trim registers.
- The trim functions in Device_cal() can be called individually in C2000Ware as ADC_setOffsetTrim(), ADC_setINLTrim() and DAC_setOffsetTrim(). These functions fetch production test trim values from M0 RAM locations, and write the values to the corresponding analog module register destinations.

Until the appropriate factory trim is loaded, the ADC and other analog modules are not specified to operate within the data sheet specifications. Similarly, if trim values other than the factory settings are placed into the trim registers, the ADC (and other modules) is not specified to operate within the data sheet specifications.

The boot ROM calls the calibration functions, so trim values are initially populated without user intervention. However, if the trims are cleared due to a module reset or modified for some other reason, then the user must call the calibration functions (defined in the C2000Ware header files).

24.12.1 ADC Zero Offset Calibration

ADC offset error is determined and calibrated during factory testing. However, the user still has the option to perform offset calibration if the end application specifically requires this. This section describes how to perform offset calibration using internal VREFLO connection for single-ended operation, and external channels for differential operation. Refer to the register descriptions for ADCCTL2.OFFTRIMMODE, ADCOFFTRIM, ADCOFFTRIM2 and ADCOFFTRIM3 for proper offset correction parameters. The C2000Ware function ADC_setMode() sets the value of OFFTRIMMODE to 1. In this mode, the ADC selects the appropriate offset calibration values from one of the three ADCOFFTRIMx registers, depending on the signal mode and channel type (odd or even) used.

Zero offset error is defined as the difference from 0 that occurs when converting a voltage at VREFLO (for single-ended operation), or the difference from MAX_CODE/2 when converting ADCINxP = ADCINxN (for differential mode). The zero offset error can be positive or negative. To correct this error, an adjustment of equal magnitude and opposite polarity is written into the ADCOFFTRIMx register. The value contained in this register is applied before the results are available in the ADC result registers. This operation is fully contained within the ADC core, so the timing of the results is not affected, and the full dynamic range of the ADC is maintained for any trim value.

Note

Regardless of the converter resolution, the size of each ADCOFFTRIMx step is (VREFHI-VREFLO)/65536.

Use the following procedure to re-calibrate the ADC offset in 12-bit single-ended mode:

1. Set ADCOFFTRIMx to +112 steps (0x70). This adds an artificial offset to account for negative offset that can reside in the ADC core.
2. Perform some multiple of 16 conversions on VREFLO (internal connection), accumulating the results (for example, 32*16 conversions = 512 conversions). Use the maximum value of ACQPS to make sure longer settling time to account for parasitic impedance of internal VREFLO connections. To sample internal VREFLO channels, use the driverlib function ADC_setupSOCRefloChannel(). After VREFLO conversion has taken place, call ADC_disableIntRefloConnection() to disconnect the ADC input multiplexer from the internal VREFLO connection.
3. Divide the accumulated result by the multiple of 16 (for example, for 512 conversions, divide by 32).
4. Set ADCOFFTRIMx to 112 – result from step 3.

Use the following procedure to recalibrate the ADC offset in 16-bit or 12-bit differential mode:

1. Set ADCOFFTRIMx to no adjustment (0x00).
2. Short ADCINxP and ADCINyN together (external connection) to a voltage near Vrefcm and accumulate some multiple of 16 conversions (for example, 32*16 conversions = 512 conversions).
3. Divide the accumulated result by the number of conversions (for example, for 512 conversions, divide by 512 for 16-bit mode, or divide by 32 for 12-bit mode).
4. Set ADCOFFTRIMx to 0 – result from step 3).

24.13 ADC Timings

The process of converting an analog voltage to a digital value is broken down into an S+H phase and a conversion phase. The ADC sample and hold circuits (S+H) are clocked by SYSCLK while the ADC conversion process is clocked by ADCCLK. ADCCLK is generated by dividing down SYSCLK based on the PRESCALE field in the ADCCTL2 register.

The S+H duration is the value of the ACQPS field of the SOC being converted, plus one, times the SYSCLK period. The user must make sure that this duration exceeds both 1 ADCCLK period and the minimum S+H duration specified in the data sheet. The conversion time is approximately 10.5 ADCCLK cycles in 12-bit mode and 29.5 ADCCLK cycles in 16-bit mode. The exact conversion time is always a whole number of SYSCLK cycles. See the timing diagrams and tables in Section 24.13.1 for exact timings.

24.13.1 ADC Timing Diagrams

The following diagrams show the ADC conversion timings for two SOCs given the following assumptions:

- SOC0 and SOC1 are configured to use the same trigger.
- No other SOCs are converting or pending when the trigger occurs.
- The round robin pointer is in a state that causes SOC0 to convert first.
- ADCINTSEL is configured to set an ADCINT flag upon end of conversion for SOC0 (whether this flag propagates through to the CPU to cause an interrupt is determined by the configurations in the Interrupt Controller module).

Table 24-11 describes the parameters in the following timing diagrams. Table 24-12.

Table 24-11. ADC Timing Parameter Descriptions

Parameter	Description
t_{SH}	The duration of the S+H window. At the end of this window, the value on the S+H capacitor becomes the voltage to be converted into a digital value. The duration is given by (ACQPS + 1) SYSCLK cycles. ACQPS can be configured individually for each SOC, so t_{SH} is not necessarily the same for different SOCs. **Note:** The value on the S+H capacitor is captured approximately 5ns before the end of the S+H window regardless of device clock settings.
t_{LAT}	The time from the end of the S+H window until the ADC results latch in the ADCRESULTx register. If the ADCRESULTx register is read before this time, the previous conversion results are returned.

Table 24-11. ADC Timing Parameter Descriptions (continued)

Parameter	Description
t_{EOC}	The time from the end of the S+H window until the S+H window for the next ADC conversion can begin. In 16-bit mode, this coincides with the latching of the conversion results, while in 12-bit mode, the subsequent sample can start before the conversion results are latched.
t_{INT}	The time from the end of the S+H window until an ADCINT flag is set (if configured). If the INTPULSEPOS bit in the ADCCTL1 register is set, t_{INT} coincides with the end of conversion (EOC) signal. If the INTPULSEPOS bit is 0, and the OFFSET field in the ADCINTCYCLE register is not 0, then there is a delay of OFFSET SYSCLK cycles before the ADCINT flag is set. This delay can be used to enter the ISR or trigger the RTDMA exactly when the sample is ready. If the INTPULSEPOS bit is 0, t_{INT} coincides with the end of the S+H window. If t_{INT} triggers a read of the ADC result register (directly through RTDMA or indirectly by triggering an ISR that reads the result), care must be taken to make sure the read occurs after the results latch (otherwise, the previous results are read).
t_{DMA}	The time from the end of the S+H window until a RTDMA read of the ADC conversion result is triggered, when ADCCTL1.TDMAEN = 1. If TDMAEN is set to 0, then the RTDMA trigger occurs at T_{INT}. In certain conditions, the ADCINT flag can be set before the ADCRESULT value is latched. To make sure that the RTDMA read occurs after the ADCRESULT value has been latched, write 1 to ADCCTL1.TDMAEN to enable RTDMA timings.

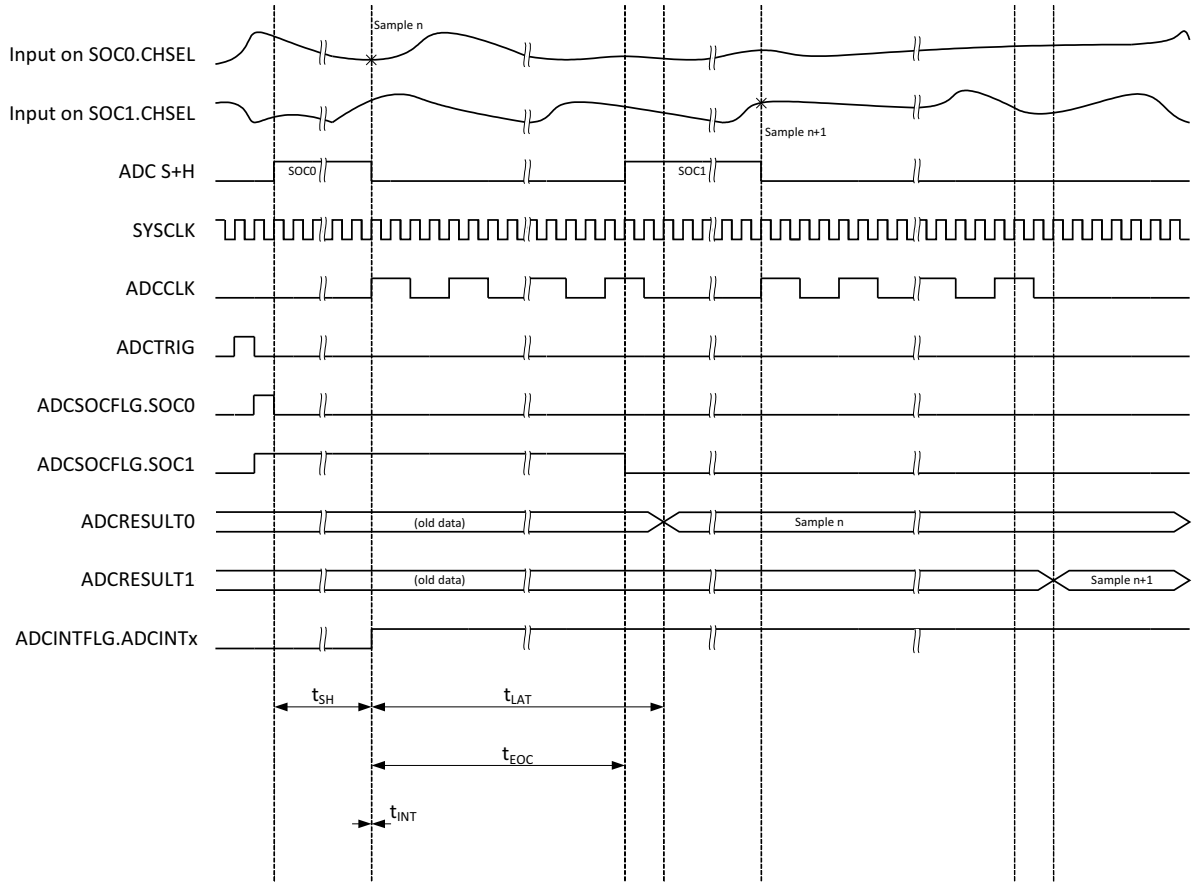


Figure 24-32. ADC Timings for 12-bit Mode in Early Interrupt Mode

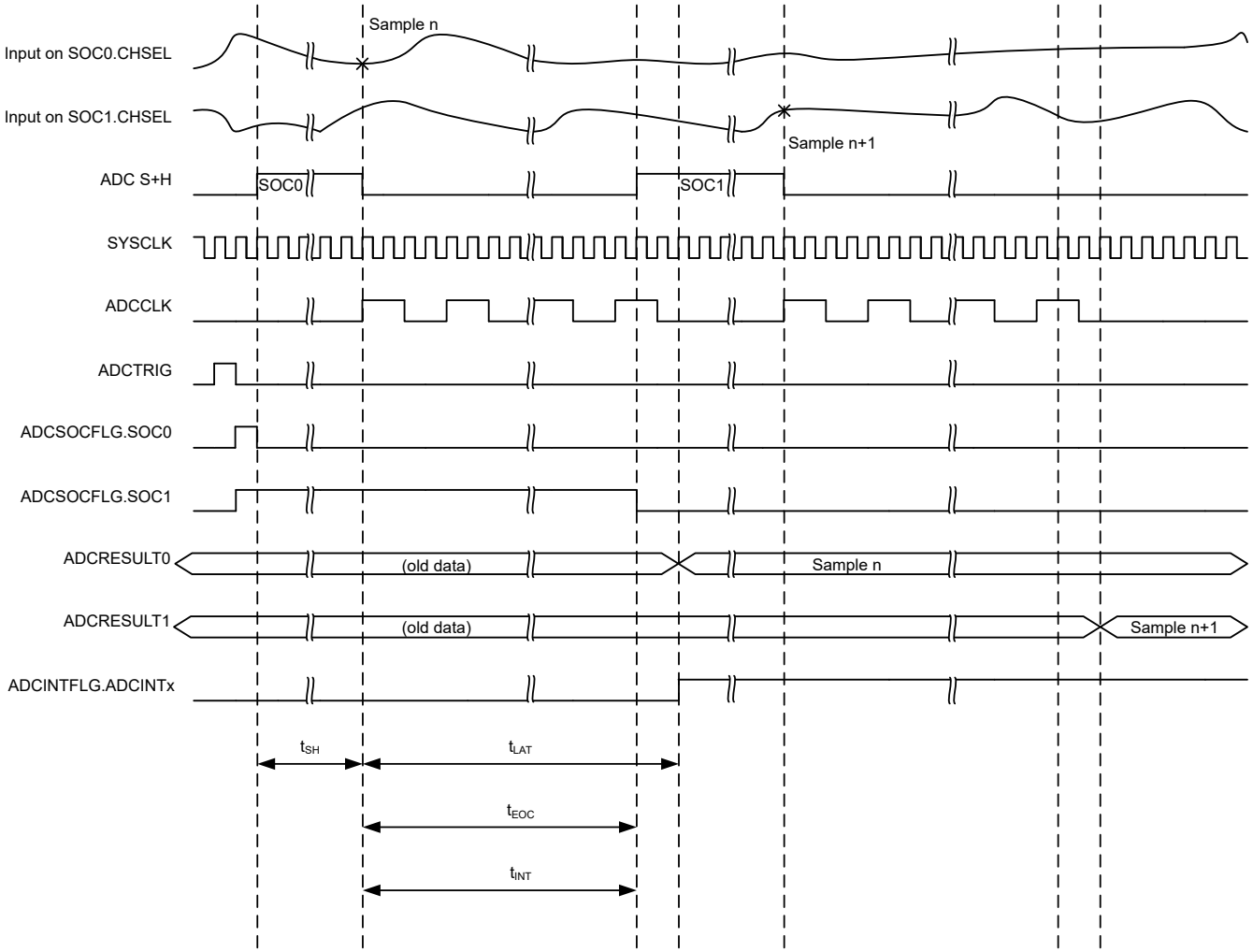


Figure 24-33. ADC Timings for 12-bit Mode in Late Interrupt Mode

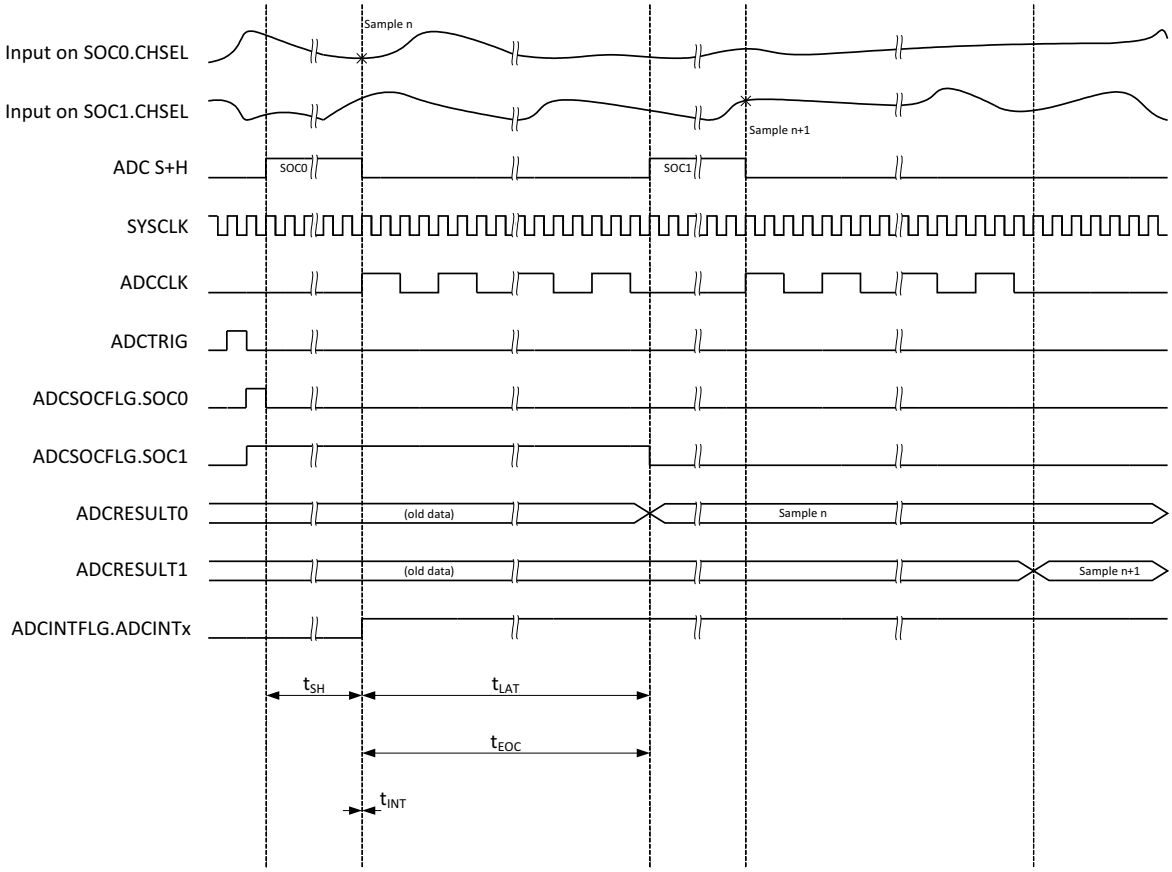


Figure 24-34. ADC Timings for 16-bit Mode in Early Interrupt Mode

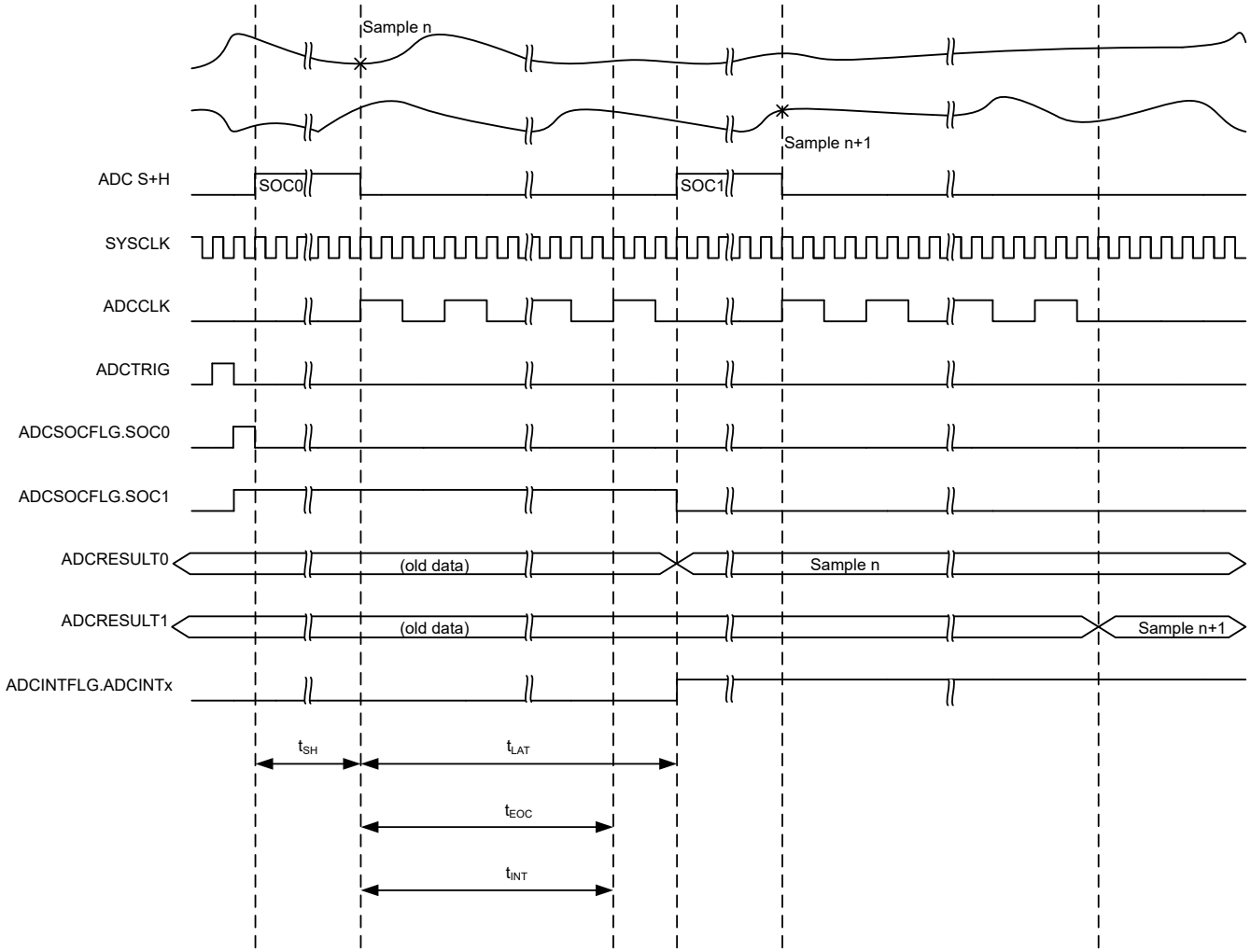


Figure 24-35. ADC Timings for 16-bit Mode in Late Interrupt Mode (SYSCLK Cycles)

Table 24-12. ADC Timings in 12-bit Mode

ADCCLK Prescale		SYSCLK Cycles				
ADCCTL2. PRESCALE	Prescale Ratio	t_{EOC}	t_{LAT}	t_{INT} (Early)(1)	t_{INT} (Late)	t_{DMA}
0	1	11	13	0	11	13
2	2	21	23	0	21	23
3	2.5	26	28	0	26	28
4	3	31	34	0	31	34
5	3.5	36	39	0	36	39
6	4	41	44	0	41	44
7	4.5	46	49	0	46	49
8	5	51	55	0	51	55
9	5.5	56	60	0	56	60
10	6	61	65	0	61	65
11	6.5	66	70	0	66	70
12	7	71	76	0	71	76
13	7.5	76	81	0	76	81
14	8	81	86	0	81	86
15	8.5	86	91	0	86	91

(1) By default, t_{INT} occurs one SYSCLK cycle after the S+H window, if INTPULSEPOS is 0. This can be changed by writing to the OFFSET field in the ADCINTCYCLE register.

Table 24-13. ADC Timings in 16-bit Mode

ADCCLK Prescale		SYSCLK Cycles				
ADCCTL2. PRESCALE	Prescale Ratio	t_{EOC}	t_{LAT}	t_{INT} (Early)(1)	t_{INT} (Late)	t_{DMA}
0	1	31	32	0	31	32
2	2	60	61	0	60	61
3	2.5	75	75	0	75	75
4	3	90	91	0	90	91
5	3.5	104	106	0	104	106
6	4	119	120	0	119	120
7	4.5	134	134	0	134	134
8	5	149	150	0	149	150
9	5.5	163	165	0	163	165
10	6	178	179	0	178	179
11	6.5	193	193	0	193	193
12	7	208	209	0	208	209
13	7.5	222	224	0	222	224
14	8	237	238	0	237	238
15	8.5	252	252	0	252	252

(1) By default, t_{INT} occurs one SYSCLK cycle after the S+H window, if INTPULSEPOS is 0. This can be changed by writing to the OFFSET field in the ADCINTCYCLE register.

24.13.2 Post-Processing Block Timings

The value of ADCRESULT is always available at time t_{LAT}, as specified in Section 24.13.1. The value of ADCPPBxRESULT, and the limit and zero-crossing comparisons are available 1 SYSCLK cycle later, as long as multiple PPB instances do not point to the same SOC. Table 24-14 shows PPB result availability timings when there are no SOCs shared between multiple PPBs. In cases where multiple PPBs point to the same SOC, PPB results become available sequentially, starting with the lowest numbered PPB. The first ADCPPBxRESULT becomes available 1 SYSCLK cycle after t_{LAT}, and each subsequent ADCPPBxRESULT becomes available 1 SYSCLK cycle after the previous PPB has completed the limit and zero-crossing comparison. The serialized PPB results are therefore spaced 2 or 3 SYSCLK cycles apart, depending on whether the comparison uses ADCPPBxRESULT or PSUM/SUM respectively. This timing is as shown in Table 24-15. PPB aggregation values (PSUM, SUM, PCOUNT, COUNT, PMAX, MAX, PMIN, MIN, PMINI, MINI, PMAXI, MAXI) are available 1 cycle after the associated ADCPPBxRESULT becomes available.

Furthermore, the LIMIT and zero-crossing compares occur 1 cycle after the compared results become available. In the case that the comparison is done against ADCPPBxRESULT, the comparison occurs 1 SYSCLK cycle after ADCPPBxRESULT becomes available. In the case that the comparison is done against PSUM or SUM, the comparison occurs 2 SYSCLK cycles after ADCPPBxRESULT is available.

Table 24-14. PPB Result Timings (One PPB per SOC)

Register	Description	Results Available
ADCRESULTy	ADC result	t_{LAT}
ADCPPBxRESULT	PPB result	t_{LAT} + 1 SYSCLK
ADCPPBxPSUM	Oversampling partial sum	t_{LAT} + 2 SYSCLK
ADCPPBxSUM	Oversampling sum	t_{LAT} + 2 SYSCLK
ADCPPBxPCOUNT	Oversampling partial sample count	t_{LAT} + 2 SYSCLK
ADCPPBxCOUNT	Oversampling sample count	t_{LAT} + 2 SYSCLK
ADCPPBxPMAX	Partial max	t_{LAT} + 2 SYSCLK
ADCPPBxMAX	Final max	t_{LAT} + 2 SYSCLK
ADCPPBxPMAXI	Partial index of max	t_{LAT} + 2 SYSCLK
ADCPPBxMAXI	Final index of max	t_{LAT} + 2 SYSCLK
ADCPPBxPMIN	Partial min	t_{LAT} + 2 SYSCLK
ADCPPBxMIN	Final min	t_{LAT} + 2 SYSCLK
ADCPPBxPMINI	Partial index of max	t_{LAT} + 2 SYSCLK
ADCPPBxMINI	Final index of max	t_{LAT} + 2 SYSCLK
Comparison	Limit and zero-crossing comparison (if using PPBxRESULT)	t_{LAT} + 2 SYSCLK
Comparison	Limit and zero-crossing comparison (if using PSUM or SUM)	t_{LAT} + 3 SYSCLK

Table 24-15. PPB Result Timings (Multiple PPBs Configured to Same SOC)

Register	Description	Results Available
ADCRESULTy	ADC result	t_{LAT}
ADCPPBxRESULT	PPB result	Varies (PPBs process serially)
ADCPPBxPSUM	Oversampling partial sum	ADCPPBxRESULT+ 1 SYSCLK
ADCPPBxSUM	Oversampling sum	ADCPPBxRESULT+ 1 SYSCLK
ADCPPBxPSUM	Oversampling partial sum	ADCPPBxRESULT+ 1 SYSCLK
ADCPPBxSUM	Oversampling sum	ADCPPBxRESULT+ 1 SYSCLK
ADCPPBxPMAX	Partial max	ADCPPBxRESULT+ 1 SYSCLK
ADCPPBxMAX	Final max	ADCPPBxRESULT+ 1 SYSCLK
ADCPPBxPMAXI	Partial index of max	ADCPPBxRESULT+ 1 SYSCLK
ADCPPBxMAXI	Final index of max	ADCPPBxRESULT+ 1 SYSCLK
ADCPPBxPMIN	Partial min	ADCPPBxRESULT+ 1 SYSCLK
ADCPPBxMIN	Final min	ADCPPBxRESULT+ 1 SYSCLK
ADCPPBxPMINI	Partial index of max	ADCPPBxRESULT+ 1 SYSCLK
ADCPPBxMINI	Final index of max	ADCPPBxRESULT+ 1 SYSCLK
Comparison	Limit and zero-crossing comparison (if using PPBxRESULT)	t_{LAT} + 2 SYSCLK
Comparison	Limit and zero-crossing comparison (if using PSUM or SUM)	t_{LAT} + 3 SYSCLK

24.14 Additional Information

The following sections contain additional practical information.

24.14.1 Ensuring Synchronous Operation

For best performance, all ADCs on the device must be operated synchronously. The device data sheet specifies the performance in both synchronous and asynchronous mode for those parameters which differ between the modes of operation.

To make sure synchronous operation, all ADCs on the device must operate in lockstep. This is accomplished by writing configurations to all ADCs that cause the sampling and conversion phases of all ADCs to be exactly aligned. The easiest way to accomplish this is to write identical values to the SOC configurations for each ADC for trigger select and ACQPS (S+H duration). In addition, synchronous ADCs must also configure identical values for the SOC priority control, burst mode, burst trigger, and burst size.

24.14.1.1 Basic Synchronous Operation

The following example configures two SOCs each on ADCA and ADCC with identical trigger select and ACQPS values. This results in synchronous operation between ADCA and ADCC. For devices with more than two ADCs, the same principles can be used to synchronize all the ADCs.

```
Example: Basic Synchronous Operation
AdcaRegs.ADCSOC0CTL.bit.CHSEL = 4;      //SOC0 converts ADCINA4
AdcaRegs.ADCSOC0CTL.bit.ACQPS = 19;     //SOC0 uses sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 10;   //SOC0 begins conversion on ePWM3 SOCB
AdccRegs.ADCSOC0CTL.bit.CHSEL = 0;      //SOC0 converts ADCINC0
AdccRegs.ADCSOC0CTL.bit.ACQPS = 19;     //SOC0 uses sample duration of 20 SYSCLK cycles
AdccRegs.ADCSOC0CTL.bit.TRIGSEL = 10;   //SOC0 begins conversion on ePWM3 SOCB

AdcaRegs.ADCSOC1CTL.bit.CHSEL = 4;      //SOC1 converts ADCINA4
AdcaRegs.ADCSOC1CTL.bit.ACQPS = 30;     //SOC1 uses sample duration of 31 SYSCLK cycles
AdcaRegs.ADCSOC1CTL.bit.TRIGSEL = 10;   //SOC1 begins conversion on ePWM3 SOCB
AdccRegs.ADCSOC1CTL.bit.CHSEL = 1;      //SOC1 converts ADCINC1
AdccRegs.ADCSOC1CTL.bit.ACQPS = 30;     //SOC1 uses sample duration of 31 SYSCLK cycles
AdccRegs.ADCSOC1CTL.bit.TRIGSEL = 10;   //SOC1 begins conversion on ePWM3 SOCB
```

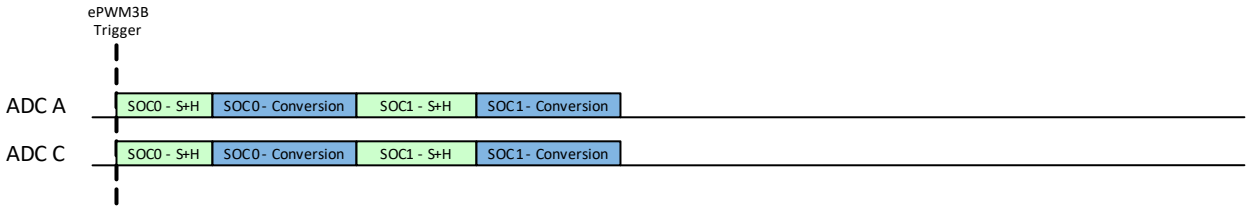


Figure 24-36. Example: Basic Synchronous Operation

Several things can be noted from Figure 24-36. First, while the ACQPS values must be the same for SOCs with the same number, different ACQPS values can be used for SOCs with different numbers. Because of this, synchronous operation does not require a single global S+H time, but instead only channels sampled simultaneously require identical S+H durations. Another important point from this example is that any channel select value can be used for any SOC. Finally, this example assumes round-robin operation. If high-priority SOCs are to be used, the priority must be configured the same on all ADCs.

24.14.1.2 Synchronous Operation with Multiple Trigger Sources

As long as each set of SOCs has identical trigger select and ACQPS settings, multiple trigger sources can be used while still achieving synchronous operation.

The following example demonstrates synchronous operation between ADCA and ADCC while using three SOCs and two trigger sources. Figure 24-37 demonstrates that any combination of relative trigger timings still results in synchronous operation.

```
Example: Synchronous Operation With Multiple Trigger Sources
AdcaRegs.ADCSOC0CTL.bit.CHSEL = 4;      //SOC0 converts ADCINA4
AdcaRegs.ADCSOC0CTL.bit.ACQPS = 19;     //SOC0 uses sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 10;   //SOC0 begins conversion on ePWM3 SOCB
AdccRegs.ADCSOC0CTL.bit.CHSEL = 0;      //SOC0 converts ADCINC0
AdccRegs.ADCSOC0CTL.bit.ACQPS = 19;     //SOC0 uses sample duration of 20 SYSCLK cycles
AdccRegs.ADCSOC0CTL.bit.TRIGSEL = 10;   //SOC0 begins conversion on ePWM3 SOCB

AdcaRegs.ADCSOC1CTL.bit.CHSEL = 4;      //SOC1 converts ADCINA4
AdcaRegs.ADCSOC1CTL.bit.ACQPS = 30;     //SOC1 uses sample duration of 31 SYSCLK cycles
AdcaRegs.ADCSOC1CTL.bit.TRIGSEL = 10;   //SOC1 begins conversion on ePWM3 SOCB
AdccRegs.ADCSOC1CTL.bit.CHSEL = 1;      //SOC1 converts ADCINC1
AdccRegs.ADCSOC1CTL.bit.ACQPS = 30;     //SOC1 uses sample duration of 31 SYSCLK cycles
AdccRegs.ADCSOC1CTL.bit.TRIGSEL = 10;   //SOC1 begins conversion on ePWM3 SOCB

AdcaRegs.ADCSOC2CTL.bit.CHSEL = 0;      //SOC2 converts ADCINA0
AdcaRegs.ADCSOC2CTL.bit.ACQPS = 19;     //SOC2 uses sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC2CTL.bit.TRIGSEL = 2;    //SOC2 begins conversion on CPU Timer1
AdccRegs.ADCSOC2CTL.bit.CHSEL = 2;      //SOC2 converts ADCINC2
AdccRegs.ADCSOC2CTL.bit.ACQPS = 19;     //SOC2 uses sample duration of 20 SYSCLK cycles
AdccRegs.ADCSOC2CTL.bit.TRIGSEL = 2;    //SOC2 begins conversion on CPU Timer1
```

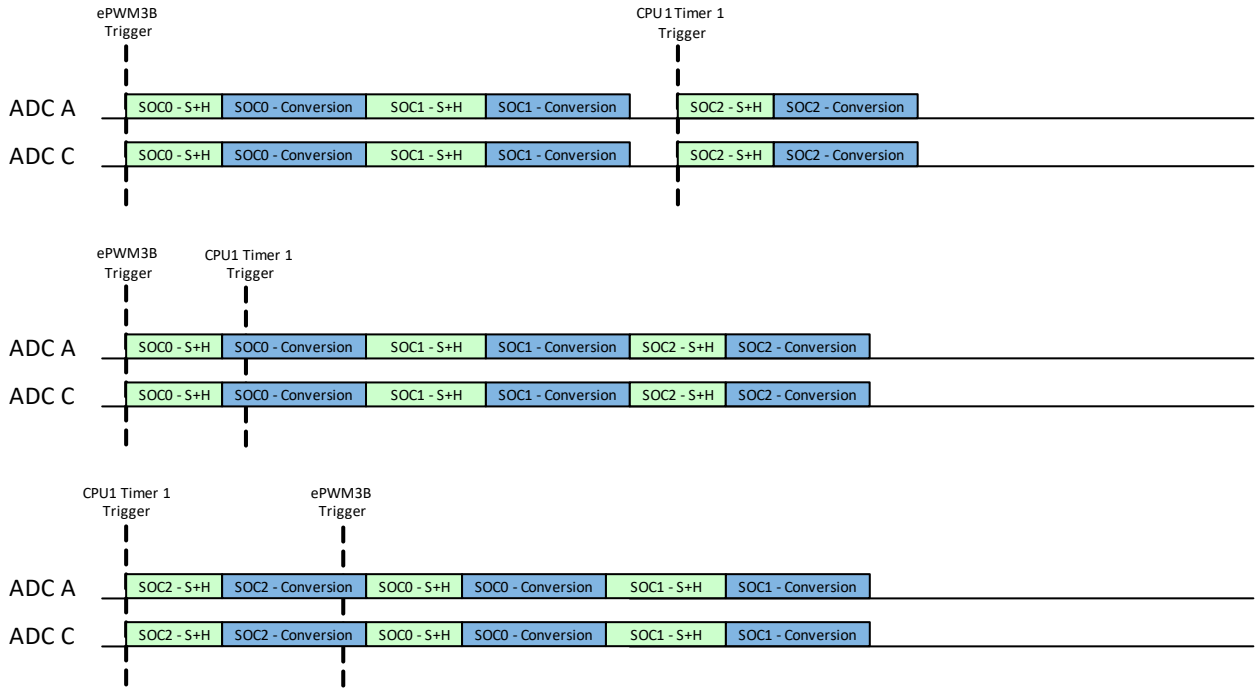


Figure 24-37. Example: Synchronous Operation with Multiple Trigger Sources

Note that any trigger source that can be selected in the TRIGSEL field can be used except for software triggering. There is no way to issue the software triggers for all ADCs simultaneously, so likely results in asynchronous operation. ADCINT1 or ADCINT2 can also be used as a trigger as long as the ADCINTSOCSEL1 and ADCINTSOCSEL2 registers are configured identically for all ADCs and software triggering is not used to start the chain of conversions.

24.14.1.3 Synchronous Operation with Uneven SOC Numbers

If only one trigger source is used, one ADC can use more SOCs than the other ADCs while still operating synchronously.

```
Example: Synchronous Operation With Uneven SOC Numbers
AdcaRegs.ADCSOC0CTL.bit.CHSEL = 4;      //SOC0 converts ADCINA4
AdcaRegs.ADCSOC0CTL.bit.ACQPS = 19;     //SOC0 uses sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 10;   //SOC0 begins conversion on ePWM3 SOCB
AdccRegs.ADCSOC0CTL.bit.CHSEL = 0;      //SOC0 converts ADCINC0
AdccRegs.ADCSOC0CTL.bit.ACQPS = 19;     //SOC0 uses sample duration of 20 SYSCLK cycles
AdccRegs.ADCSOC0CTL.bit.TRIGSEL = 10;   //SOC0 begins conversion on ePWM3 SOCB

AdcaRegs.ADCSOC1CTL.bit.CHSEL = 4;      //SOC1 converts ADCINA4
AdcaRegs.ADCSOC1CTL.bit.ACQPS = 30;     //SOC1 uses sample duration of 31 SYSCLK cycles
AdcaRegs.ADCSOC1CTL.bit.TRIGSEL = 10;   //SOC1 begins conversion on ePWM3 SOCB
AdccRegs.ADCSOC1CTL.bit.CHSEL = 1;      //SOC1 converts ADCINC1
AdccRegs.ADCSOC1CTL.bit.ACQPS = 30;     //SOC1 uses sample duration of 31 SYSCLK cycles
AdccRegs.ADCSOC1CTL.bit.TRIGSEL = 10;   //SOC1 begins conversion on ePWM3 SOCB

AdcaRegs.ADCSOC2CTL.bit.CHSEL = 0;      //SOC2 converts ADCINA0
AdcaRegs.ADCSOC2CTL.bit.ACQPS = 30;     //SOC2 uses sample duration of 31 SYSCLK cycles
AdcaRegs.ADCSOC2CTL.bit.TRIGSEL = 10;   //SOC2 begins conversion on ePWM3 SOCB
```

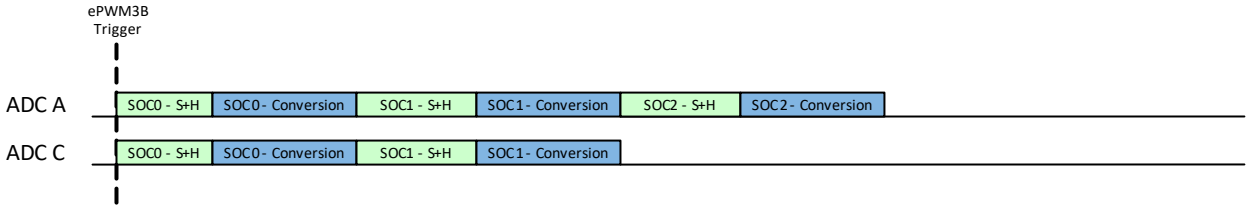


Figure 24-38. Example: Synchronous Operation with Uneven SOC Numbers

Note that if the trigger comes again before all SOCs have completed the conversions, ADCC begins converting immediately on SOC0 while ADCA does not start converting SOC0 again until SOC2 is complete. This results in asynchronous operation, so care must be taken to not overflow the trigger.

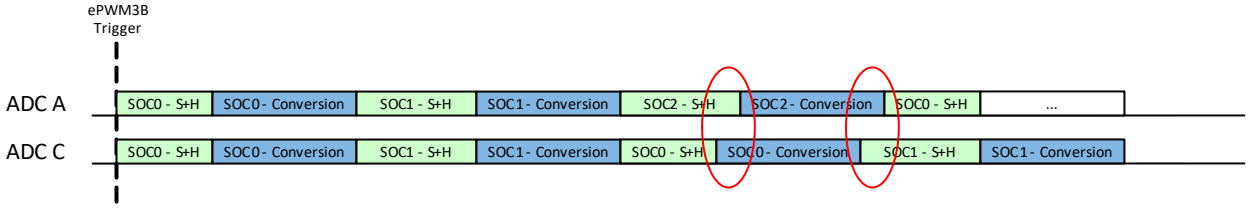


Figure 24-39. Example: Asynchronous Operation with Uneven SOC Numbers – Trigger Overflow

24.14.1.4 Synchronous Operation with Different Resolutions

Configuring different ADCs to use different resolutions results in asynchronous operation. This occurs because the conversion time for 12-bit mode and 16-bit mode are different. Synchronous operation requires both the start and end of the conversion phase to be aligned, so even using the same S+H window duration does not result in synchronous operation.

```
Example: Asynchronous Operation with Different Resolutions
//ADCA = 12-bit mode
AdcaRegs.ADCSOC0CTL.bit.CHSEL = 4;      //SOC0 converts ADCINA4
AdcaRegs.ADCSOC0CTL.bit.ACQPS = 50;     //SOC0 uses sample duration of 51 SYSCLK cycles
AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 10;   //SOC0 begins conversion on ePWM3 SOCB

//ADCB = 16-bit mode
AdcbRegs.ADCSOC0CTL.bit.CHSEL = 0;      //SOC0 converts ADCINB0/B1
AdcbRegs.ADCSOC0CTL.bit.ACQPS = 50;     //SOC0 uses sample duration of 51 SYSCLK cycles
AdcbRegs.ADCSOC0CTL.bit.TRIGSEL = 10;   //SOC0 begins conversion on ePWM3 SOCB
```

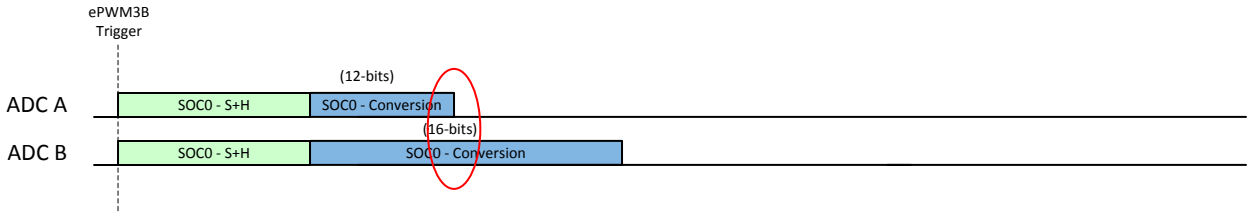


Figure 24-40. Example: Asynchronous Operation with Different Resolutions

To achieve synchronous operation while using both 12-bit and 16-bit resolution, conversions must be done in parallel at one resolution. Once conversions are complete at one resolution, the CPU must switch the resolution on all ADCs and then cause another trigger (this trigger must not be a software SOC force, as all ADCs cannot be started simultaneously using this method).

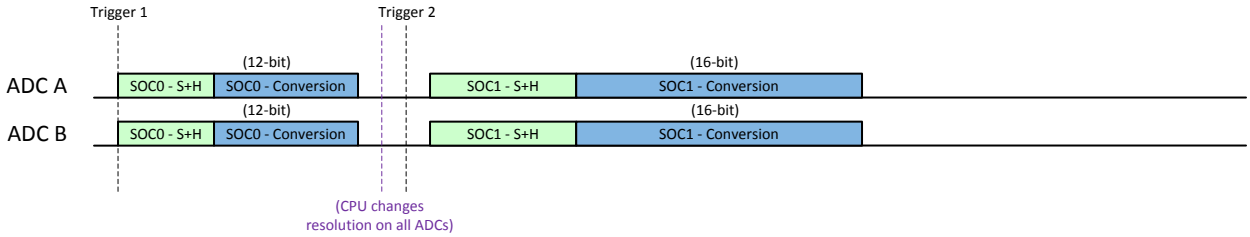


Figure 24-41. Example: Synchronous Operation with Different Resolutions

24.14.1.5 Non-overlapping Conversions

If conversion timings can be made sure to not overlap by the user, then it is not necessary to configure all SOCs identically on all ADCs to achieve performance equivalent to synchronous operation. For example, if the two ADC triggers in a system come from two ePWM sources that are always 180-degrees out-of-phase, then SOC0 can be used for both ADCA and ADCC with different trigger sources and different ACQPS values.

```
Example: Operation with Non-overlapping Conversions
//ePWM3 SOCA and SOCB are 180 degrees out of phase
AdcaRegs.ADCSOC0CTL.bit.CHSEL = 4;      //SOC0 converts ADCINA4
AdcaRegs.ADCSOC0CTL.bit.ACQPS = 19;     //SOC0 uses sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 10;   //SOC0 begins conversion on ePWM3 SOCB
AdccRegs.ADCSOC0CTL.bit.CHSEL = 0;      //SOC0 converts ADCINC0
AdccRegs.ADCSOC0CTL.bit.ACQPS = 19;     //SOC0 uses sample duration of 20 SYSCLK cycles
AdccRegs.ADCSOC0CTL.bit.TRIGSEL = 9;    //SOC0 begins conversion on ePWM3 SOCA
```

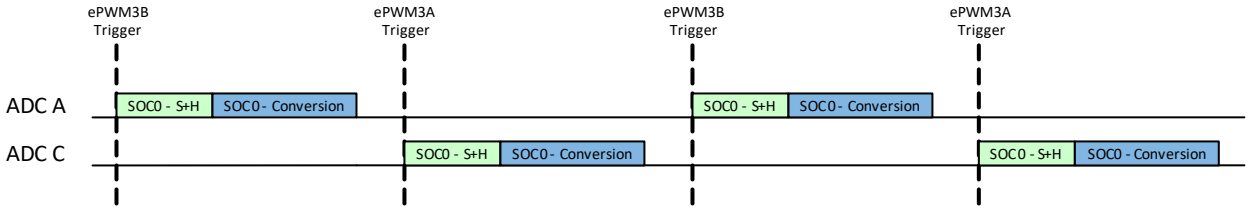


Figure 24-42. Example: Synchronous Equivalent Operation with Non-Overlapping Conversions

24.14.2 Choosing an Acquisition Window Duration

For correct operation, the input signal to the ADC must be allowed adequate time to charge the sample and hold capacitor, Ch. Typically, the S+H duration is chosen such that the sampling capacitor is charged to within ½ LSB or ¼ LSB of the final value, depending on the tolerable settling error.

The best methodology to determine the required settling time is to simulate the ADC and ADC driving circuits to make sure adequate settling performance. See *ADC Input Circuit Evaluation for C2000 MCUs* and *Charge-Sharing Driving Circuits for C2000 ADCs* for additional guidance on ADC signal conditioning circuit design and evaluation.

An approximation of the required settling time can also be determined using an RC settling model. The time constant for the model is given by the equation:

$$\tau = (R_S + R_{on}) \times C_h + R_S \times \left(C_S + C_p\right) \tag{15}$$

And the number of time constants needed is given by the equation:

$$k = ln\left(\frac{2^n}{settling\ error}\right) - ln\left(\frac{C_S + C_P}{CH}\right) \tag{16}$$

So the total S+H time must be set to at least:

$$t = k \cdot \tau \tag{17}$$

Where the following parameters are provided by the ADC input model in the device data sheet:

- n = ADC resolution (in bits)
- R_{ON} = ADC sampling switch resistance (provided in Ω)
- C_H = ADC sampling capacitor (provided in pF)
- C_p = ADC channel parasitic input capacitance (provided in pF)

And the following parameters are dependent on the application design:

- settling error = tolerable settling error (in LSBs)
- R_s = ADC driving circuit source impedance (typically in Ω or kΩ)
- C_S = capacitance on ADC input pin (typically in pF or nF)

For example, assuming the following parameters:

- n = 12-bits
- R_{ON} = 500Ω
- C_H = 12.5pF
- C_p = 12.7pF
- settling error = ¼ LSB
- R_s = 180Ω
- C_s = 150pF

The time constant is calculated as:

$$\tau = (180\Omega + 500\Omega) \times 12.5pF + 180\Omega \times (150pF + 12.7pF) = 37.8ns \quad (18)$$

And the number of required time constants is:

$$k = ln\left(\frac{2^{12}}{0.25}\right) - ln\left(\frac{150pF + 12.7pF}{12.5pF}\right) = 9.70 - 2.57 = 7.13 \quad (19)$$

So the S+H time must be set to at least: 37.8ns × 7.13 = 270ns

If SYSCLK = 200MHz , then each SYSCLK cycle is 5ns. S+H duration is 270ns/5ns = 54 SYSCLK cycles, so ACQPS for this input is set to at least CEILING(54.0) – 1 = 53.

While this gives a rough estimate of the required acquisition window, a better method is to setup a circuit with the ADC input model, a model of the source impedance/capacitance, and any board parasitics in SPICE (or similar software) and simulate to verify that the sampling capacitor settles to the desired accuracy.

Note

The device data sheet specifies a minimum ADC S+H window duration. Do not use an ACQPS value that gives a duration less than this specification.

24.14.3 Achieving Simultaneous Sampling

While each ADC does not have dual S+H circuits, achieving simultaneous sampling is accomplished by setting the SOC triggers on two or more ADC modules to use the same trigger source. The following example demonstrates simultaneous sampling on 3 ADCs based on an ePWM3 event. ADCINA3, ADCINB2, and ADCINC5 are sampled. An acquisition window of 20 SYSCLK cycles is used, but different durations are possible. The following example assumes all ADCs are configured for 12-bit mode operation.

```
AdcaRegs.ADCSOC0CTL.bit.CHSEL = 3;        //SOC0 converts ADCINA3
AdcaRegs.ADCSOC0CTL.bit.ACQPS = 19;       //SOC0 uses sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 10;     //SOC0 begins conversion on ePWM3 SOCB
AdcbRegs.ADCSOC0CTL.bit.CHSEL = 2;        //SOC0 converts ADCINB2
AdcbRegs.ADCSOC0CTL.bit.ACQPS = 19;       //SOC0 uses sample duration of 20 SYSCLK cycles
AdcbRegs.ADCSOC0CTL.bit.TRIGSEL = 10;     //SOC0 begins conversion on ePWM3 SOCB
AdccRegs.ADCSOC0CTL.bit.CHSEL = 5;        //SOC0 converts ADCINC5
AdccRegs.ADCSOC0CTL.bit.ACQPS = 19;       //SOC0 uses sample duration of 20 SYSCLK cycles
AdccRegs.ADCSOC0CTL.bit.TRIGSEL = 10;     //SOC0 begins conversion on ePWM3 SOCB
```

When the ePWM3 trigger is received, all 3 ADCs begin converting in parallel immediately. All results are stored in the ADCRESULT0 register for each ADC. Note that this assumes that all ADCs are idle when the trigger is received. If one or more ADCs is busy, the samples do not happen at exactly the same time.

24.14.4 Result Register Mapping

The ADC results and the ADC PPB results are duplicated for each memory bus controller in the system. Bus controllers include all C29x cores and RTDMAs present on the specific part family and part number. For each bus controller, no access configuration is needed to allow read access to the result registers, and no contention occurs in cases where multiple bus controllers try to read the ADC results simultaneously.

24.14.5 Internal Temperature Sensor

The internal temperature sensor measures the junction temperature of the device. The output of the sensor can be sampled with the ADC through an internal connection. This can be enabled on channel ADCIN20 on ADCA, ADCIN20 on ADCC, and on the CMPSS3_HP3 and CMPSS5_HP3 inputs by setting the ENABLE bit in the TSNSCTL register.

To convert the temperature sensor reading into a temperature, pass the temperature sensor reading to the ADC_getTemperatureC() function in the ADC driverlib.

Note

To sample the temperature sensor, the ADC must be in single-ended 12-bit mode.

If the temperature sensor is sampled in 16-bit mode, this can cause incorrect ADC results.

24.14.6 Designing an External Reference Circuit

Figure 24-43 shows the basic organization of the external voltage reference generation circuitry. TI recommends that a single reference voltage generation source is shared by all ADC modules. This minimizes reference voltage mismatch between ADC modules. For best performance, the externally generated reference voltage must be buffered by a precision op-amp with good bandwidth and low output impedance before being driven into the ADC reference pin. A capacitor between the high and low reference pins must be placed on the PCB as close to the pins as practical to help absorb high-frequency currents. A series resistor (typically <1Ω) in series with this capacitor is sometimes necessary to maintain op-amp stability.

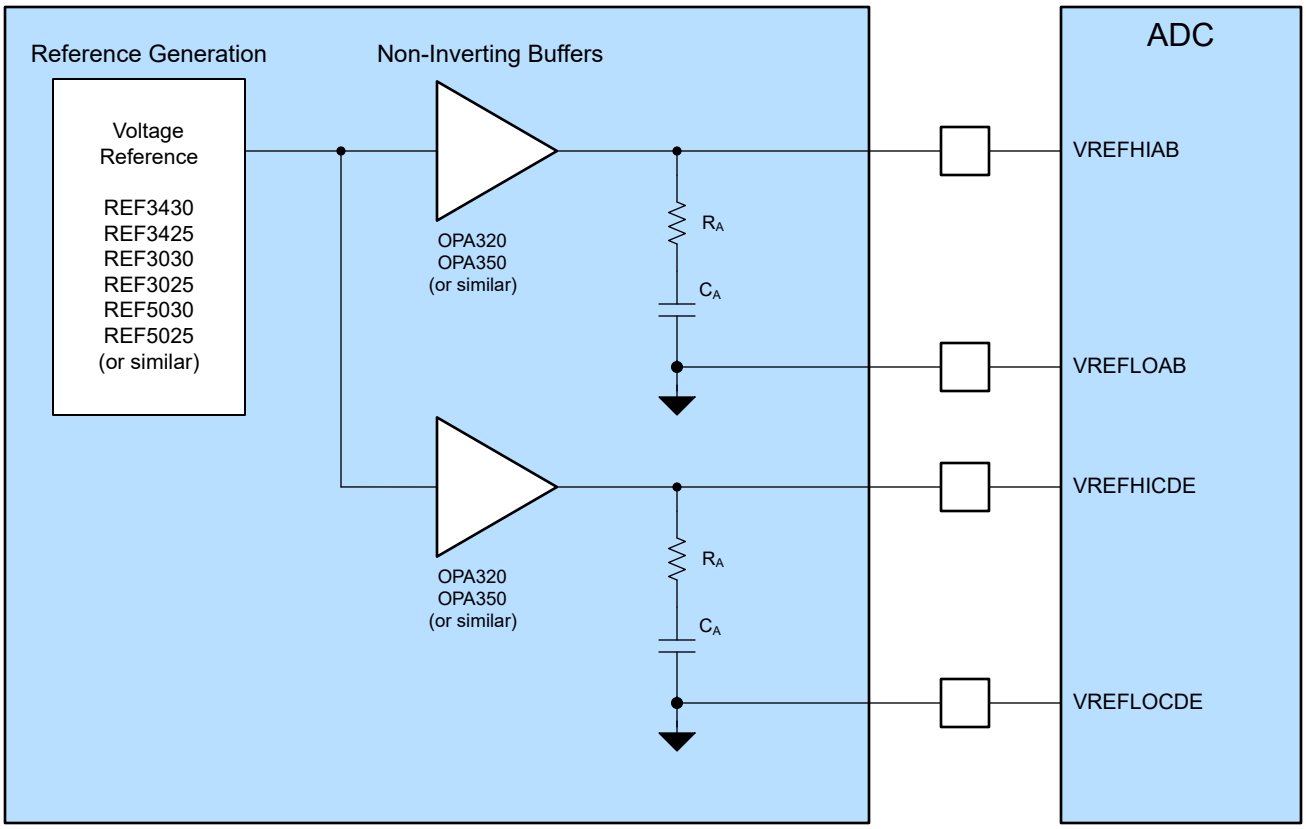


Figure 24-43. ADC Reference System

24.14.7 Internal Test Mode

For diagnostic purposes, the ADC can sample various internal node voltages using a special input selection mux called INTERNALTEST. The INTERNALTEST mux can be sampled on the ADCIN23 input channel of any of the ADC modules. Additionally, redundant INTERNALTEST connections are available on ADC-A channel ADCIN16, and on ADC-B channel ADCIN18. Internal test mode can be used to sample the VDDCORE voltage, VREFLO, VDDA, VSSA, and the CMPSS DAC outputs.

To enable internal test mode, write the desired node selection to the TESTSEL field of the INTERNALTESTCTL analog subsystem register. For details of internal test mux connections to various internal device voltage nodes, refer to the INTERNALTESTCTL register description.

To disable internal test mode, write 0 to the TESTSEL field of the INTERNALTESTCTL register.

When using internal test mode, the following special considerations apply:

- The minimum sampling window size (ACQPS) when converting INTERNALTEST is 2.56µs (512 SYSCLK cycles at 200MHz SYSCLK) .
- The effective resolution of the CMPSS DAC outputs to INTERNALTEST is 6 bits.

For more information on the CMPSS module and how to configure the CMPSS DAC, see the *Comparator Subsystem (CMPSS)* chapter.

Note

For functional safety and to prevent unintended writes to the INTERNALTESTCTL register, write 1 to CONFIGLOCK.LOCKBIT in ANALOG_SUBSYS_REGS. When set, this bit prevents any further writes to the INTERNALTESTCTL register from taking effect until the next device reset.

24.14.8 ADC Gain and Offset Calibration

Using a dedicated input channel, gain balancing between multiple ADCs can be performed. By calculating the relative gain error between the ADCs, conversion results can be post-processed in software such that each ADC has the same output for each equivalent input.

To perform gain calibration, configure the ADC SOC to sample channel ADCIN21. On this channel, the voltage sampled by all ADCs when triggered is (VREFHI * 0.9).

ADC offset can also be calibrated by using dedicated input channels to measure the reference low voltage. VREFLOAB can be measured by sampling channel ADCIN22 on ADC-C, ADC-D, and ADC-E. VREFLOCDE can be measured on ADC-A and ADC-B by sampling channel ADCIN17 for the odd channel, and ADCIN22 for the even channel.

24.15 Software

24.15.1 ADC Registers to Driverlib Functions

Table 24-16. ADC Registers to Driverlib Functions

File	Driverlib Function
ADCCTL1	
adc.h	ADC_setInterruptPulseMode
adc.h	ADC_enableAltDMATiming
adc.h	ADC_disableAltDMATiming
adc.h	ADC_enableExtMuxPreselect
adc.h	ADC_disableExtMuxPreselect
adc.h	ADC_enableConverter
adc.h	ADC_disableConverter
adc.h	ADC_isBusy
ADCCTL2	
adc.c	ADC_setMode
adc.c	ADC_setINLTrim
adc.h	ADC_setPrescaler
adc.h	ADC_selectOffsetTrimMode
ADCBURSTCTL	
adc.h	ADC_setBurstModeConfig
adc.h	ADC_enableBurstMode
adc.h	ADC_disableBurstMode
ADCINTFLG	
adc.h	ADC_getIntResultStatus
adc.h	ADC_getInterruptStatus
adc.h	ADC_clearInterruptStatus
ADCINTFLGCLR	
adc.h	ADC_clearInterruptStatus
ADCINTOVF	
adc.h	ADC_getInterruptOverflowStatus
adc.h	ADC_clearInterruptOverflowStatus
ADCINTOVFCLR	
adc.h	ADC_clearInterruptOverflowStatus
ADCINTSEL1N2	
adc.h	ADC_enableInterrupt
adc.h	ADC_disableInterrupt
adc.h	ADC_setInterruptSource
adc.h	ADC_enableContinuousMode
adc.h	ADC_disableContinuousMode
ADCINTSEL3N4	
-	See INTSEL1N2
ADCSOCPRICTL	
adc.h	ADC_setSOCPriority
ADCINTSOCSEL1	
adc.h	ADC_setInterruptSOCTrigger
ADCINTSOCSEL2	

Table 24-16. ADC Registers to Driverlib Functions (continued)

File	Driverlib Function
adc.h	ADC_setInterruptSOCTrigger
ADCSOCFLG1	
-	
ADCSOCFRC1	
adc.h	ADC_forceSOC
adc.h	ADC_forceMultipleSOC
ADCSOCOVF1	
-	
ADCSOCOVFCLR1	
-	
ADCSOC0CTL	
adc.h	ADC_setupSOC
adc.h	ADC_selectSOCExtChannel
ADCSOC1CTL	
-	See SOC0CTL
ADCSOC2CTL	
-	See SOC0CTL
ADCSOC3CTL	
-	See SOC0CTL
ADCSOC4CTL	
-	See SOC0CTL
ADCSOC5CTL	
-	See SOC0CTL
ADCSOC6CTL	
-	See SOC0CTL
ADCSOC7CTL	
-	See SOC0CTL
ADCSOC8CTL	
-	See SOC0CTL
ADCSOC9CTL	
-	See SOC0CTL
ADCSOC10CTL	
-	See SOC0CTL
ADCSOC11CTL	
-	See SOC0CTL
ADCSOC12CTL	
-	See SOC0CTL
ADCSOC13CTL	
-	See SOC0CTL
ADCSOC14CTL	
-	See SOC0CTL
ADCSOC15CTL	
-	See SOC0CTL
ADCSOC16CTL	
-	

Table 24-16. ADC Registers to Driverlib Functions (continued)

File	Driverlib Function
ADCSOC17CTL	
-	
ADCSOC18CTL	
-	
ADCSOC19CTL	
-	
ADCSOC20CTL	
-	
ADCSOC21CTL	
-	
ADCSOC22CTL	
-	
ADCSOC23CTL	
-	
ADCSOC24CTL	
-	
ADCSOC25CTL	
-	
ADCSOC26CTL	
-	
ADCSOC27CTL	
-	
ADCSOC28CTL	
-	
ADCSOC29CTL	
-	
ADCSOC30CTL	
-	
ADCSOC31CTL	
-	
ADCEVTSTAT	
adc.h	ADC_getPPBEventStatus
ADCEVTCLR	
adc.h	ADC_clearPPBEventStatus
ADCEVTSEL	
adc.h	ADC_enablePPBEvent
adc.h	ADC_disablePPBEvent
ADCEVTINTSEL	
adc.h	ADC_enablePPBEventInterrupt
adc.h	ADC_disablePPBEventInterrupt
ADCOSDETECT	
adc.h	ADC_configOSDetectMode
ADCCOUNTER	
-	
ADCREV	

Table 24-16. ADC Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
ADCOFFTRIM	
-	
ADCOFFTRIM2	
-	
ADCOFFTRIM3	
-	
ADCPPB1CONFIG	
adc.h	ADC_setupPPB
adc.h	ADC_enablePPBEventCBCClear
adc.h	ADC_disablePPBEventCBCClear
adc.h	ADC_enablePPBAbsoluteValue
adc.h	ADC_disablePPBAbsoluteValue
adc.h	ADC_setPPBShiftValue
adc.h	ADC_selectPPBSyncInput
adc.h	ADC_forcePPBSync
adc.h	ADC_selectPPBOSINTSource
adc.h	ADC_selectPPBCompareSource
adc.h	ADC_enablePPBTwosComplement
adc.h	ADC_disablePPBTwosComplement
adc.h	ADC_disablePPBExtendedLowLimit
ADCPPB1STAMP	
adc.h	ADC_getPPBDelayTimeStamp
ADCPPB1OFFCAL	
adc.h	ADC_setPPBCalibrationOffset
ADCPPB1OFFREF	
adc.h	ADC_setPPBReferenceOffset
ADCPPB1TRIPHI	
adc.c	ADC_setPPBTripLimits
ADCPPB1TRIPLO	
adc.c	ADC_setPPBTripLimits
adc.h	ADC_enablePPBExtendedLowLimit
ADCPPBTRIP1FILCTL	
-	
ADCPPBTRIP1FILCLKCTL	
-	
ADCPPB2CONFIG	
-	See PPB1CONFIG
ADCPPB2STAMP	
-	See PPB1STAMP
ADCPPB2OFFCAL	
-	See PPB1OFFCAL
ADCPPB2OFFREF	
-	See PPB1OFFREF
ADCPPB2TRIPHI	

Table 24-16. ADC Registers to Driverlib Functions (continued)

File	Driverlib Function
-	See PPB1TRIPHI
ADCPPB2TRIPLO	
-	See PPB1TRIPLO
ADCPPBTRIP2FILCTL	
-	
ADCPPBTRIP2FILCLKCTL	
-	
ADCPPB3CONFIG	
-	See PPB1CONFIG
ADCPPB3STAMP	
-	See PPB1STAMP
ADCPPB3OFFCAL	
-	See PPB1OFFCAL
ADCPPB3OFFREF	
-	See PPB1OFFREF
ADCPPB3TRIPHI	
-	See PPB1TRIPHI
ADCPPB3TRIPLO	
-	See PPB1TRIPLO
ADCPPBTRIP3FILCTL	
-	
ADCPPBTRIP3FILCLKCTL	
-	
ADCPPB4CONFIG	
-	See PPB1CONFIG
ADCPPB4STAMP	
-	See PPB1STAMP
ADCPPB4OFFCAL	
-	See PPB1OFFCAL
ADCPPB4OFFREF	
-	See PPB1OFFREF
ADCPPB4TRIPHI	
-	See PPB1TRIPHI
ADCPPB4TRIPLO	
-	See PPB1TRIPLO
ADCPPBTRIP4FILCTL	
-	
ADCPPBTRIP4FILCLKCTL	
-	
ADCSAFECHECKRESEN	
adc.h	ADC_configSOCSafetyCheckerInput
ADCSAFECHECKRESEN2	
adc.h	ADC_configSOCSafetyCheckerInput
ADCINTCYCLE	
adc.h	ADC_setInterruptCycleOffset

Table 24-16. ADC Registers to Driverlib Functions (continued)

File	Driverlib Function
ADCINLTRIM1	
adc.c	ADC_setINLTrim
ADCINLTRIM2	
adc.c	ADC_setINLTrim
ADCINLTRIM3	
-	
ADCINLTRIM4	
adc.c	ADC_setINLTrim
ADCINLTRIM5	
adc.c	ADC_setINLTrim
ADCINLTRIM6	
-	
ADCREV2	
-	
ADCREP1CTL	
adc.c	ADC_configureRepeater
adc.h	ADC_getRepeaterStatus
adc.h	ADC_triggerRepeaterMode
adc.h	ADC_triggerRepeaterActiveMode
adc.h	ADC_triggerRepeaterModuleBusy
adc.h	ADC_triggerRepeaterSelect
adc.h	ADC_triggerRepeaterSyncIn
adc.h	ADC_forceRepeaterTriggerSync
ADCREP1N	
adc.c	ADC_configureRepeater
adc.h	ADC_triggerRepeaterCount
ADCREP1PHASE	
adc.c	ADC_configureRepeater
adc.h	ADC_triggerRepeaterPhase
ADCREP1SPREAD	
adc.c	ADC_configureRepeater
adc.h	ADC_triggerRepeaterSpread
ADCREP1FRC	
adc.h	ADC_forceRepeaterTrigger
ADCREP2CTL	
-	
ADCREP2N	
-	
ADCREP2PHASE	
-	
ADCREP2SPREAD	
-	
ADCREP2FRC	
-	
ADCPPB1LIMIT	

Table 24-16. ADC Registers to Driverlib Functions (continued)

File	Driverlib Function
adc.h	ADC_setPPBCountLimit
adc.h	ADC_getPPBCountLimit
ADCPPBP1PCOUNT	
-	
ADCPPB1CONFIG2	
adc.h	ADC_setPPBShiftValue
adc.h	ADC_selectPPBSyncInput
adc.h	ADC_forcePPBSync
adc.h	ADC_selectPPBOSINTSource
adc.h	ADC_selectPPBCompareSource
ADCPPB1PSUM	
-	
ADCPPB1PMAX	
-	
ADCPPB1PMAXI	
-	
ADCPPB1PMIN	
-	
ADCPPB1PMINI	
-	
ADCPPB1TRIPLO2	
adc.c	ADC_setPPBTripLimits
ADCPPB2LIMIT	
-	
ADCPPBP2PCOUNT	
-	
ADCPPB2CONFIG2	
-	
ADCPPB2PSUM	
-	
ADCPPB2PMAX	
-	
ADCPPB2PMAXI	
-	
ADCPPB2PMIN	
-	
ADCPPB2PMINI	
-	
ADCPPB2TRIPLO2	
-	
ADCPPB3LIMIT	
-	
ADCPPBP3PCOUNT	
-	
ADCPPB3CONFIG2	

Table 24-16. ADC Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
ADCPPB3PSUM	
-	
ADCPPB3PMAX	
-	
ADCPPB3PMAXI	
-	
ADCPPB3PMIN	
-	
ADCPPB3PMINI	
-	
ADCPPB3TRIPLO2	
-	
ADCPPB4LIMIT	
-	
ADCPPBP4PCOUNT	
-	
ADCPPB4CONFIG2	
-	
ADCPPB4PSUM	
-	
ADCPPB4PMAX	
-	
ADCPPB4PMAXI	
-	
ADCPPB4PMIN	
-	
ADCPPB4PMINI	
-	
ADCPPB4TRIPLO2	
-	
ADCRESULT0	
adc.h	ADC_readResult
ADCRESULT1	
-	See RESULT0
ADCRESULT2	
-	See RESULT0
ADCRESULT3	
-	See RESULT0
ADCRESULT4	
-	See RESULT0
ADCRESULT5	
-	See RESULT0
ADCRESULT6	
-	See RESULT0

Table 24-16. ADC Registers to Driverlib Functions (continued)

File	Driverlib Function
ADCRESULT7	
-	See RESULT0
ADCRESULT8	
-	See RESULT0
ADCRESULT9	
-	See RESULT0
ADCRESULT10	
-	See RESULT0
ADCRESULT11	
-	See RESULT0
ADCRESULT12	
-	See RESULT0
ADCRESULT13	
-	See RESULT0
ADCRESULT14	
-	See RESULT0
ADCRESULT15	
-	See RESULT0
ADCRESULT16	
-	
ADCRESULT17	
-	
ADCRESULT18	
-	
ADCRESULT19	
-	
ADCRESULT20	
-	
ADCRESULT21	
-	
ADCRESULT22	
-	
ADCRESULT23	
-	
ADCRESULT24	
-	
ADCRESULT25	
-	
ADCRESULT26	
-	
ADCRESULT27	
-	
ADCRESULT28	
-	
ADCRESULT29	

Table 24-16. ADC Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
ADCRESULT30	
-	
ADCRESULT31	
-	
ADCPPB1RESULT	
adc.h	ADC_readPPBResult
ADCPPB2RESULT	
-	See PPB1RESULT
ADCPPB3RESULT	
-	See PPB1RESULT
ADCPPB4RESULT	
-	See PPB1RESULT
ADCPPB1SUM	
-	
ADCPPB1COUNT	
-	
ADCPPB2SUM	
-	
ADCPPB2COUNT	
-	
ADCPPB3SUM	
-	
ADCPPB3COUNT	
-	
ADCPPB4SUM	
-	
ADCPPB4COUNT	
-	
ADCPPB1MAX	
-	
ADCPPB1MAXI	
-	
ADCPPB1MIN	
-	
ADCPPB1MINI	
-	
ADCPPB2MAX	
-	
ADCPPB2MAXI	
-	
ADCPPB2MIN	
-	
ADCPPB2MINI	
-	

Table 24-16. ADC Registers to Driverlib Functions (continued)

File	Driverlib Function
ADCPPB3MAX	
-	
ADCPPB3MAXI	
-	
ADCPPB3MIN	
-	
ADCPPB3MINI	
-	
ADCPPB4MAX	
-	
ADCPPB4MAXI	
-	
ADCPPB4MIN	
-	
ADCPPB4MINI	
-	
ADCCHECKCONFIG	
adc.h	ADC_enableSafetyChecker
adc.h	ADC_disableSafetyChecker
adc.h	ADC_forceSafetyCheckerSync
ADCCHECKSTATUS	
adc.h	ADC_getSafetyCheckerStatus
ADCRESSEL1	
adc.h	ADC_configureSafetyChecker
ADCRESSEL2	
-	
ADCTOLERANCE	
adc.h	ADC_setSafetyCheckerTolerance
ADCCHECKRESULT1	
adc.h	ADC_getSafetyCheckerResult
ADCCHECKRESULT2	
-	
ADCOOTFLG	
adc.h	ADC_getSafetyCheckStatus
adc.h	ADC_clearSafetyCheckStatus
ADCOOTFLGCLR	
adc.h	ADC_clearSafetyCheckStatus
ADCRES1OVF	
-	
ADCRES1OVFCLR	
-	
ADCRES2OVF	
-	
ADCRES2OVFCLR	
-	

Table 24-16. ADC Registers to Driverlib Functions (continued)

File	Driverlib Function
ADCCHECKINTFLG	
adc.h	ADC_getSafetyCheckIntStatus
adc.h	ADC_clearSafetyCheckIntStatus
ADCCHECKINTFLGCLR	
adc.h	ADC_clearSafetyCheckIntStatus
ADCCHECKINTSEL1	
adc.h	ADC_enableSafetyCheckInt
adc.h	ADC_disableSafetyCheckInt
ADCCHECKINTSEL2	
-	
ADCCHECKINTSEL3	
-	
ADCCHECKEVT1SEL1	
adc.h	ADC_enableSafetyCheckEvt
adc.h	ADC_disableSafetyCheckEvt
ADCCHECKEVT1SEL2	
-	
ADCCHECKEVT1SEL3	
-	
ADCCHECKEVT2SEL1	
-	
ADCCHECKEVT2SEL2	
-	
ADCCHECKEVT2SEL3	
-	
ADCCHECKEVT3SEL1	
-	
ADCCHECKEVT3SEL2	
-	
ADCCHECKEVT3SEL3	
-	
ADCCHECKEVT4SEL1	
-	
ADCCHECKEVT4SEL2	
-	
ADCCHECKEVT4SEL3	
-	

24.15.2 ADC Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/adc

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

24.15.2.1 ADC Software Triggering - SINGLE_CORE

FILE: adc_ex1_soc_software.c

This example converts some voltages on ADCA and ADCC based on a software trigger.

The ADCC will not convert until ADCA is complete, so the ADCs will not run asynchronously. However, this is much less efficient than allowing the ADCs to convert synchronously in parallel (for example, by using an ePWM trigger).

External Connections

- A0, A1, C2, and C3 should be connected to signals to convert

Watch Variables

- *myADC0Result0* - Digital representation of the voltage on pin A0
- *myADC0Result1* - Digital representation of the voltage on pin A1
- *myADC1Result0* - Digital representation of the voltage on pin C2
- *myADC1Result1* - Digital representation of the voltage on pin C3

24.15.2.2 ADC ePWM Triggering - SINGLE_CORE

FILE: adc_ex2_soc_epwm.c

This example sets up ePWM1 to periodically trigger a conversion on ADCA.

External Connections

- A0 should be connected to a signal to convert

Watch Variables

- *myADC0Results* - A sequence of analog-to-digital conversion samples from pin A0. The time between samples is determined based on the period of the ePWM timer.

24.15.2.3 ADC Temperature Sensor Conversion - SINGLE_CORE

FILE: adc_ex3_temp_sensor.c

This example sets up the ePWM to periodically trigger the ADC. The ADC converts the internal connection to the temperature sensor, which is then interpreted as a temperature by calling the ADC_getTemperatureC() function.

Watch Variables

- *sensorSample* - The raw reading from the temperature sensor
- *sensorTemp* - The interpretation of the sensor sample as a temperature in degrees Celsius.

24.15.2.4 ADC Synchronous SOC Software Force (adc_soc_software_sync) - SINGLE_CORE

FILE: adc_ex4_software_sync.c

This example converts some voltages on ADCA and ADCC using input 5 of the input X-BAR as a software force. Input 5 is triggered by toggling GPIO0, but any spare GPIO could be used. This method will ensure that both ADCs start converting at exactly the same time.

External Connections

- A0, A1, C0, C1 pins should be connected to signals to convert

Watch Variables

- *myADC0Result0* : a digital representation of the voltage on pin A0
- *myADC0Result1* : a digital representation of the voltage on pin A1

- *myADC1Result0* : a digital representation of the voltage on pin C0
- *myADC1Result1* : a digital representation of the voltage on pin C1

24.15.2.5 ADC Continuous Triggering (adc_soc_continuous) - SINGLE_CORE

FILE: adc_ex5_soc_continuous.c

This example sets up the ADC to convert continuously, achieving maximum sampling rate.
External Connections
- A0 pin should be connected to signal to convert

Watch Variables
- *adcAResults* - A sequence of analog-to-digital conversion samples from pin A0. The time between samples is the minimum possible based on the ADC speed.

24.15.2.6 ADC Continuous Conversions Read by DMA (adc_soc_continuous_dma) - SINGLE_CORE

FILE: adc_ex6_continuous_dma.c

This example sets up two ADC channels to convert simultaneously. The results will be transferred by the DMA into a buffer in RAM.

External Connections
- A3 & C3 pins should be connected to signals to convert

Watch Variables
- *myADC0DataBuffer* : a digital representation of the voltage on pin A3
- *myADC1DataBuffer* : a digital representation of the voltage on pin C3

24.15.2.7 ADC PPB Offset (adc_ppb_offset) - SINGLE_CORE

FILE: adc_ex7_ppb_offset.c

This example software triggers the ADC. Some SOCs have automatic offset adjustment applied by the post-processing block. After the program runs, the memory will contain ADC & post-processing block(PPB) results.

External Connections
- A0, C0 pins should be connected to signals to convert

Watch Variables
- *myADC0Result* : a digital representation of the voltage on pin A0
- *myADC0PPBResult* : a digital representation of the voltage on pin A0, minus 100 LSBs of automatically added offset
- *myADC1Result* : a digital representation of the voltage on pin C0
- *myADC1PPBResult* : a digital representation of the voltage on pin C0 plus 100 LSBs of automatically added offset

24.15.2.8 ADC PPB Limits (adc_ppb_limits) - SINGLE_CORE

FILE: adc_ex8_ppb_limits.c

This example sets up the ePWM to periodically trigger the ADC. If the results are outside of the defined range, the post-processing block will generate an interrupt.

The default limits are 1000LSBs and 3000LSBs. With VREFHI set to 3.3V, the PPB will generate an interrupt if the input voltage goes above about 2.4V or below about 0.8V.

External Connections
- A0 should be connected to a signal to convert

Watch Variables
- None

24.15.2.9 ADC PPB Delay Capture (adc_ppb_delay) - SINGLE_CORE

FILE: adc_ex9_ppb_delay.c

This example demonstrates delay capture using the post-processing block.

Two asynchronous ADC triggers are setup:
- ePWM1, with period 2048, triggering SOC0 to convert on pin A0
- ePWM2, with period 9999, triggering SOC1 to convert on pin A1
  Each conversion generates an ISR at the end of the conversion. In the ISR for SOC0, a conversion counter is incremented and the PPB is checked to determine if the sample was delayed.
  After the program runs, the memory will contain:

conversion : the sequence of conversions using SOC0 that were delayed
- *delay* : the corresponding delay of each of the delayed conversions

24.15.2.10 ADC ePWM Triggering Multiple SOC - SINGLE_CORE

FILE: adc_ex10_multiple_soc_epwm.c

This example sets up ePWM1 to periodically trigger a set of conversions on ADCA and ADCC. This example demonstrates multiple ADCs working together to process of a batch of conversions using the available parallelism across multiple ADCs.

ADCA Interrupt ISRs are used to read results of both ADCA and ADCC.

External Connections
- A0, A1, A2 and C2, C3, C4 pins should be connected to signals to be converted.

Watch Variables
- *adcAResult0* - Digital representation of the voltage on pin A0
- *adcAResult1* - Digital representation of the voltage on pin A1
- *adcAResult2* - Digital representation of the voltage on pin A2
- *adcCResult0* - Digital representation of the voltage on pin C2
- *adcCResult1* - Digital representation of the voltage on pin C3
- *adcCResult2* - Digital representation of the voltage on pin C4

24.15.2.11 ADC Burst Mode - SINGLE_CORE

FILE: adc_ex11_burst_mode_epwm.c

This example sets up ePWM1 to periodically trigger ADCA using burst mode. This allows for different channels to be sampled with each burst.

Each burst triggers 3 conversions. A0 and A1 are part of every burst while the third conversion rotates between A2, A3, and A4. This allows high importance signals to be sampled at high speed while lower priority signals can be sampled at a lower rate.

ADCA Interrupt ISRs are used to read results for ADCA.

External Connections
- A0, A1, A2, A3, A4

Watch Variables
- *adcAResult0* - Digital representation of the voltage on pin A0
- *adcAResult1* - Digital representation of the voltage on pin A1
- *adcAResult2* - Digital representation of the voltage on pin A2
- *adcAResult3* - Digital representation of the voltage on pin A3
- *adcAResult4* - Digital representation of the voltage on pin A4

24.15.2.12 ADC Burst Mode Oversampling - SINGLE_CORE

FILE: adc_ex12_burst_mode_oversampling.c

This example is an ADC oversampling example implemented with software. The ADC SOCs are configured in burst mode, triggered by the ePWM SOC A event trigger.

External Connection

- A2

Watch Variables

- *lv_results* - Array of digital values measured on pin A2 (oversampling is configured by Oversampling_Amount)

24.15.2.13 ADC SOC Oversampling - SINGLE_CORE

FILE: adc_ex13_soc_oversampling.c

This example sets up ePWM1 to periodically trigger a set of conversions on ADCA including multiple SOCs that all convert A2 to achieve oversampling on A2.

ADCA Interrupt ISRs are used to read results of ADCA.

External Connections

- A0, A1, A2 should be connected to signals to be converted.

Watch Variables

- *adcAResult0* - Digital representation of the voltage on pin A0
- *adcAResult1* - Digital representation of the voltage on pin A1
- *adcAResult2* - Digital representation of the voltage on pin A2

24.15.2.14 ADC PPB PWM trip (adc_ppb_pwm_trip) - SINGLE_CORE

FILE: adc_ex14_ppb_pwm_trip.c

This example demonstrates EPWM tripping through ADC limit detection PPB block. ADCAINT1 is configured to periodically trigger the ADCA channel 2 post initial software forced trigger. The limit detection post-processing block(PPB) is configured and if the ADC results are outside of the defined range, the post-processing block will generate an ADCxEVTy event. This event is configured as EPWM trip source through configuring EPWM XBAR and corresponding EPWM's trip zone and digital compare sub-modules. The example showcases

- one-shot
- cycle-by-cycle
- and direct tripping of PWMs through ADCAEVT1 source via Digital compare submodule.

The default limits are 0LSBs and 3600LSBs. With VREFHI set to 3.3V, the PPB will generate a trip event if the input voltage goes above about 2.9V.

External Connections

- A2 should be connected to a signal to convert
- Observe the following signals on an oscilloscope
  - ePWM1(GPIO0 - GPIO1)
  - ePWM2(GPIO2 - GPIO3)
  - ePWM3(GPIO4 - GPIO5)
-

Watch Variables

- adcA2Results - digital representation of the voltage on pin A2

24.15.2.15 ADC Trigger Repeater Oversampling - SINGLE_CORE

FILE: adc_ex15_trigger_repeater_oversampling.c

This example configures ADC for oversampling using trigger repeater block. The ePWM1 is configured to periodically trigger the ADC SOC and the trigger repeater module is configured to generate 4 repeated pulses.

Post-processing block will take the repeated pulses, accumulates them and stores the results in ppb sum register.

External Connections

- A0 should be connected to signals to convert.

Watch Variables

- *myADC0Result* - Digital representation of the voltage on pin A0
- *myPPB0Result* - Digital representation of the voltage of the 4 repeated pulses on pin A0

24.15.2.16 ADC Trigger Repeater Undersampling - SINGLE_CORE

FILE: adc_ex16_trigger_repeater_undersampling.c

This example configures ADC for undersampling using trigger repeater block. The ePWM1 is configured to periodically trigger the ADC SOC and the trigger repeater module is configured to generate 1 pulse out of three pulses. Post-processing block will take the undersampled pulse, accumulates them and stores the results in ppb sum register.

External Connections

- A0 should be connected to signals to convert.

Watch Variables

- *myADC0Result* - Digital representation of the voltage on pin A0
- *myPPB0Result* - Digital representation of the voltage of the undersample pulse on pin A0

24.15.2.17 ADC Safety Checker - SINGLE_CORE

FILE: adc_ex17_safety_checker.c

This example compares the absolute value of the two ADC conversion results with the set tolerance value.

The ADCA is used with A0 and A1 to compare the two ADC conversions. If the difference between two conversion results exceeds the value configured as tolerance then the ADC result safety checker tile generates an interrupt signal from out-of-tolerance.

External Connections

- A0 and A1 should be connected to signals to convert.

Watch Variables

- *myADC0Result0* - Digital representation of the voltage on pin A0
- *myADC0Result1* - Digital representation of the voltage on pin A1
- *tolerance* - Set digital tolerance limit for ADC safety checker
- *count* - number of times the OOT flag is generated

24.16 ADC Registers

This section describes the ADC Registers.

24.16.1 ADC Base Address Table

Table 24-17. ADC Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
ADC_RESULT_REGS	ADCARESULT_BASE	0x303C_0000	-	YES	YES	YES	YES	YES	-	YES
ADC_RESULT_REGS	ADCBRESULT_BASE	0x303C_1000	-	YES	YES	YES	YES	YES	-	YES
ADC_RESULT_REGS	ADCCRESULT_BASE	0x303C_2000	-	YES	YES	YES	YES	YES	-	YES
ADC_RESULT_REGS	ADCDRESULT_BASE	0x303C_3000	-	YES	YES	YES	YES	YES	-	YES
ADC_RESULT_REGS	ADCERESULT_BASE	0x303C_4000	-	YES	YES	YES	YES	YES	-	YES
ADC_REGS	ADCA_BASE	0x700A_0000	YES	YES	YES	YES	YES	YES	-	YES
ADC_REGS	ADCB_BASE	0x700A_1000	YES	YES	YES	YES	YES	YES	-	YES
ADC_REGS	ADCC_BASE	0x700A_2000	YES	YES	YES	YES	YES	YES	-	YES
ADC_REGS	ADCD_BASE	0x700A_3000	YES	YES	YES	YES	YES	YES	-	YES
ADC_REGS	ADCE_BASE	0x700A_4000	YES	YES	YES	YES	YES	YES	-	YES
ADC_SAFECHECK_REGS	ADCSAFETYCHECK1_BASE	0x700B_0000	YES	YES	YES	YES	YES	YES	-	YES
ADC_SAFECHECK_REGS	ADCSAFETYCHECK2_BASE	0x700B_1000	YES	YES	YES	YES	YES	YES	-	YES
ADC_SAFECHECK_REGS	ADCSAFETYCHECK3_BASE	0x700B_2000	YES	YES	YES	YES	YES	YES	-	YES
ADC_SAFECHECK_REGS	ADCSAFETYCHECK4_BASE	0x700B_3000	YES	YES	YES	YES	YES	YES	-	YES
ADC_SAFECHECK_REGS	ADCSAFETYCHECK5_BASE	0x700B_4000	YES	YES	YES	YES	YES	YES	-	YES
ADC_SAFECHECK_REGS	ADCSAFETYCHECK6_BASE	0x700B_5000	YES	YES	YES	YES	YES	YES	-	YES
ADC_SAFECHECK_REGS	ADCSAFETYCHECK7_BASE	0x700B_6000	YES	YES	YES	YES	YES	YES	-	YES
ADC_SAFECHECK_REGS	ADCSAFETYCHECK8_BASE	0x700B_7000	YES	YES	YES	YES	YES	YES	-	YES
ADC_SAFECHECK_REGS	ADCSAFETYCHECK9_BASE	0x700B_8000	YES	YES	YES	YES	YES	YES	-	YES
ADC_SAFECHECK_REGS	ADCSAFETYCHECK10_BASE	0x700B_9000	YES	YES	YES	YES	YES	YES	-	YES
ADC_SAFECHECK_INTEVT_REGS	ADCSAFETYCHECKINTEVT1_BASE	0x700C_0000	YES	YES	YES	YES	YES	YES	-	YES
ADC_SAFECHECK_INTEVT_REGS	ADCSAFETYCHECKINTEVT2_BASE	0x700C_1000	YES	YES	YES	YES	YES	YES	-	YES
ADC_SAFECHECK_INTEVT_REGS	ADCSAFETYCHECKINTEVT3_BASE	0x700C_2000	YES	YES	YES	YES	YES	YES	-	YES
ADC_GLOBAL_REGS	ADCGLOBAL_BASE	0x700C_8000	YES	YES	YES	YES	YES	YES	-	YES

24.16.2 ADC_RESULT_REGS Registers

Table 24-18 lists the memory-mapped registers for the ADC_RESULT_REGS registers. All register offset addresses not listed in Table 24-18 should be considered as reserved locations and the register contents should not be modified.

Table 24-18. ADC_RESULT_REGS Registers

Offset	Acronym	Register Name	Protection
0h	ADCRESULT0	ADC Result 0 Register	
2h	ADCRESULT1	ADC Result 1 Register	
4h	ADCRESULT2	ADC Result 2 Register	
6h	ADCRESULT3	ADC Result 3 Register	
8h	ADCRESULT4	ADC Result 4 Register	
Ah	ADCRESULT5	ADC Result 5 Register	
Ch	ADCRESULT6	ADC Result 6 Register	
Eh	ADCRESULT7	ADC Result 7 Register	
10h	ADCRESULT8	ADC Result 8 Register	
12h	ADCRESULT9	ADC Result 9 Register	
14h	ADCRESULT10	ADC Result 10 Register	
16h	ADCRESULT11	ADC Result 11 Register	
18h	ADCRESULT12	ADC Result 12 Register	
1Ah	ADCRESULT13	ADC Result 13 Register	
1Ch	ADCRESULT14	ADC Result 14 Register	
1Eh	ADCRESULT15	ADC Result 15 Register	
20h	ADCRESULT16	ADC Result 16 Register	
22h	ADCRESULT17	ADC Result 17 Register	
24h	ADCRESULT18	ADC Result 18 Register	
26h	ADCRESULT19	ADC Result 19 Register	
28h	ADCRESULT20	ADC Result 20 Register	
2Ah	ADCRESULT21	ADC Result 21 Register	
2Ch	ADCRESULT22	ADC Result 22 Register	
2Eh	ADCRESULT23	ADC Result 23 Register	
30h	ADCRESULT24	ADC Result 24 Register	
32h	ADCRESULT25	ADC Result 25 Register	
34h	ADCRESULT26	ADC Result 26 Register	
36h	ADCRESULT27	ADC Result 27 Register	
38h	ADCRESULT28	ADC Result 28 Register	
3Ah	ADCRESULT29	ADC Result 29 Register	
3Ch	ADCRESULT30	ADC Result 30 Register	
3Eh	ADCRESULT31	ADC Result 31 Register	
40h	ADCPPB1RESULT	ADC Post Processing Block 1 Result Register	
44h	ADCPPB2RESULT	ADC Post Processing Block 2 Result Register	
48h	ADCPPB3RESULT	ADC Post Processing Block 3 Result Register	
4Ch	ADCPPB4RESULT	ADC Post Processing Block 4 Result Register	
50h	ADCPPB1SUM	ADC PPB 1 Final Sum Result Register	
54h	ADCPPB1COUNT	ADC PPB1 Final Conversion Count Register	
58h	ADCPPB2SUM	ADC PPB 2 Final Sum Result Register	
5Ch	ADCPPB2COUNT	ADC PPB2 Final Conversion Count Register	

Table 24-18. ADC_RESULT_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
60h	ADCPPB3SUM	ADC PPB 3 Final Sum Result Register	
64h	ADCPPB3COUNT	ADC PPB3 Final Conversion Count Register	
68h	ADCPPB4SUM	ADC PPB 4 Final Sum Result Register	
6Ch	ADCPPB4COUNT	ADC PPB4 Final Conversion Count Register	
70h	ADCPPB1MAX	ADC PPB 1 Final Max Result Register	
74h	ADCPPB1MAXI	ADC PPB 1 Final Max Index Result Register	
78h	ADCPPB1MIN	ADC PPB 1 Final Min Result Register	
7Ch	ADCPPB1MINI	ADC PPB 1 Final Min Index Result Register	
80h	ADCPPB2MAX	ADC PPB 2 Final Max Result Register	
84h	ADCPPB2MAXI	ADC PPB 2 Final Max Index Result Register	
88h	ADCPPB2MIN	ADC PPB 2 Final Min Result Register	
8Ch	ADCPPB2MINI	ADC PPB 2 Final Min Index Result Register	
90h	ADCPPB3MAX	ADC PPB 3 Final Max Result Register	
94h	ADCPPB3MAXI	ADC PPB 3 Final Max Index Result Register	
98h	ADCPPB3MIN	ADC PPB 3 Final Min Result Register	
9Ch	ADCPPB3MINI	ADC PPB 3 Final Min Index Result Register	
A0h	ADCPPB4MAX	ADC PPB 4 Final Max Result Register	
A4h	ADCPPB4MAXI	ADC PPB 4 Final Max Index Result Register	
A8h	ADCPPB4MIN	ADC PPB 4 Final Min Result Register	
ACh	ADCPPB4MINI	ADC PPB 4 Final Min Index Result Register	

Complex bit access types are encoded to fit into small table cells. Table 24-19 shows the codes that are used for access types in this section.

Table 24-19. ADC_RESULT_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

24.16.2.1 ADCRESULT0 Register (Offset = 0h) [Reset = 0000h]

ADCRESULT0 is shown in Figure 24-44 and described in Table 24-20.

Return to the Summary Table.

ADC Result 0 Register

Figure 24-44. ADCRESULT0 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-20. ADCRESULT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 0 16-bit ADC result. After the ADC completes a conversion of SOC0, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.2 ADCRESULT1 Register (Offset = 2h) [Reset = 0000h]

ADCRESULT1 is shown in Figure 24-45 and described in Table 24-21.

Return to the Summary Table.

ADC Result 1 Register

Figure 24-45. ADCRESULT1 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-21. ADCRESULT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 1 16-bit ADC result. After the ADC completes a conversion of SOC1, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.3 ADCRESULT2 Register (Offset = 4h) [Reset = 0000h]

ADCRESULT2 is shown in Figure 24-46 and described in Table 24-22.

Return to the Summary Table.

ADC Result 2 Register

Figure 24-46. ADCRESULT2 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-22. ADCRESULT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 2 16-bit ADC result. After the ADC completes a conversion of SOC2, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.4 ADCRESULT3 Register (Offset = 6h) [Reset = 0000h]

ADCRESULT3 is shown in Figure 24-47 and described in Table 24-23.

Return to the Summary Table.

ADC Result 3 Register

Figure 24-47. ADCRESULT3 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-23. ADCRESULT3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 3 16-bit ADC result. After the ADC completes a conversion of SOC3, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.5 ADCRESULT4 Register (Offset = 8h) [Reset = 0000h]

ADCRESULT4 is shown in Figure 24-48 and described in Table 24-24.

Return to the Summary Table.

ADC Result 4 Register

Figure 24-48. ADCRESULT4 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-24. ADCRESULT4 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 4 16-bit ADC result. After the ADC completes a conversion of SOC4, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.6 ADCRESULT5 Register (Offset = Ah) [Reset = 0000h]

ADCRESULT5 is shown in Figure 24-49 and described in Table 24-25.

Return to the Summary Table.

ADC Result 5 Register

Figure 24-49. ADCRESULT5 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-25. ADCRESULT5 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 5 16-bit ADC result. After the ADC completes a conversion of SOC5, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.7 ADCRESULT6 Register (Offset = Ch) [Reset = 0000h]

ADCRESULT6 is shown in Figure 24-50 and described in Table 24-26.

Return to the Summary Table.

ADC Result 6 Register

Figure 24-50. ADCRESULT6 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-26. ADCRESULT6 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 6 16-bit ADC result. After the ADC completes a conversion of SOC6, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.8 ADCRESULT7 Register (Offset = Eh) [Reset = 0000h]

ADCRESULT7 is shown in Figure 24-51 and described in Table 24-27.

Return to the Summary Table.

ADC Result 7 Register

Figure 24-51. ADCRESULT7 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-27. ADCRESULT7 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 7 16-bit ADC result. After the ADC completes a conversion of SOC7, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.9 ADCRESULT8 Register (Offset = 10h) [Reset = 0000h]

ADCRESULT8 is shown in Figure 24-52 and described in Table 24-28.

Return to the Summary Table.

ADC Result 8 Register

Figure 24-52. ADCRESULT8 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-28. ADCRESULT8 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 8 16-bit ADC result. After the ADC completes a conversion of SOC8, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.10 ADCRESULT9 Register (Offset = 12h) [Reset = 0000h]

ADCRESULT9 is shown in Figure 24-53 and described in Table 24-29.

Return to the Summary Table.

ADC Result 9 Register

Figure 24-53. ADCRESULT9 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-29. ADCRESULT9 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 9 16-bit ADC result. After the ADC completes a conversion of SOC9, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.11 ADCRESULT10 Register (Offset = 14h) [Reset = 0000h]

ADCRESULT10 is shown in Figure 24-54 and described in Table 24-30.

Return to the Summary Table.

ADC Result 10 Register

Figure 24-54. ADCRESULT10 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-30. ADCRESULT10 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 10 16-bit ADC result. After the ADC completes a conversion of SOC10, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.12 ADCRESULT11 Register (Offset = 16h) [Reset = 0000h]

ADCRESULT11 is shown in Figure 24-55 and described in Table 24-31.

Return to the Summary Table.

ADC Result 11 Register

Figure 24-55. ADCRESULT11 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-31. ADCRESULT11 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 11 16-bit ADC result. After the ADC completes a conversion of SOC11, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.13 ADCRESULT12 Register (Offset = 18h) [Reset = 0000h]

ADCRESULT12 is shown in Figure 24-56 and described in Table 24-32.

Return to the Summary Table.

ADC Result 12 Register

Figure 24-56. ADCRESULT12 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-32. ADCRESULT12 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 12 16-bit ADC result. After the ADC completes a conversion of SOC12, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.14 ADCRESULT13 Register (Offset = 1Ah) [Reset = 0000h]

ADCRESULT13 is shown in Figure 24-57 and described in Table 24-33.

Return to the Summary Table.

ADC Result 13 Register

Figure 24-57. ADCRESULT13 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-33. ADCRESULT13 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 13 16-bit ADC result. After the ADC completes a conversion of SOC13, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.15 ADCRESULT14 Register (Offset = 1Ch) [Reset = 0000h]

ADCRESULT14 is shown in Figure 24-58 and described in Table 24-34.

Return to the Summary Table.

ADC Result 14 Register

Figure 24-58. ADCRESULT14 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-34. ADCRESULT14 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 14 16-bit ADC result. After the ADC completes a conversion of SOC14, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.16 ADCRESULT15 Register (Offset = 1Eh) [Reset = 0000h]

ADCRESULT15 is shown in Figure 24-59 and described in Table 24-35.

Return to the Summary Table.

ADC Result 15 Register

Figure 24-59. ADCRESULT15 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-35. ADCRESULT15 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 15 16-bit ADC result. After the ADC completes a conversion of SOC15, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.17 ADCRESULT16 Register (Offset = 20h) [Reset = 0000h]

ADCRESULT16 is shown in Figure 24-60 and described in Table 24-36.

Return to the Summary Table.

ADC Result 16 Register

Figure 24-60. ADCRESULT16 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-36. ADCRESULT16 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 16 16-bit ADC result. After the ADC completes a conversion of SOC16, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.18 ADCRESULT17 Register (Offset = 22h) [Reset = 0000h]

ADCRESULT17 is shown in Figure 24-61 and described in Table 24-37.

Return to the Summary Table.

ADC Result 17 Register

Figure 24-61. ADCRESULT17 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-37. ADCRESULT17 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 17 16-bit ADC result. After the ADC completes a conversion of SOC17, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.19 ADCRESULT18 Register (Offset = 24h) [Reset = 0000h]

ADCRESULT18 is shown in Figure 24-62 and described in Table 24-38.

Return to the Summary Table.

ADC Result 18 Register

Figure 24-62. ADCRESULT18 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-38. ADCRESULT18 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 18 16-bit ADC result. After the ADC completes a conversion of SOC18, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.20 ADCRESULT19 Register (Offset = 26h) [Reset = 0000h]

ADCRESULT19 is shown in Figure 24-63 and described in Table 24-39.

Return to the Summary Table.

ADC Result 19 Register

Figure 24-63. ADCRESULT19 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-39. ADCRESULT19 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 19 16-bit ADC result. After the ADC completes a conversion of SOC19, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.21 ADCRESULT20 Register (Offset = 28h) [Reset = 0000h]

ADCRESULT20 is shown in Figure 24-64 and described in Table 24-40.

Return to the Summary Table.

ADC Result 20 Register

Figure 24-64. ADCRESULT20 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-40. ADCRESULT20 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 20 16-bit ADC result. After the ADC completes a conversion of SOC20, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.22 ADCRESULT21 Register (Offset = 2Ah) [Reset = 0000h]

ADCRESULT21 is shown in Figure 24-65 and described in Table 24-41.

Return to the Summary Table.

ADC Result 21 Register

Figure 24-65. ADCRESULT21 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-41. ADCRESULT21 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 21 16-bit ADC result. After the ADC completes a conversion of SOC21, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.23 ADCRESULT22 Register (Offset = 2Ch) [Reset = 0000h]

ADCRESULT22 is shown in Figure 24-66 and described in Table 24-42.

Return to the Summary Table.

ADC Result 22 Register

Figure 24-66. ADCRESULT22 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-42. ADCRESULT22 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 22 16-bit ADC result. After the ADC completes a conversion of SOC22, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.24 ADCRESULT23 Register (Offset = 2Eh) [Reset = 0000h]

ADCRESULT23 is shown in Figure 24-67 and described in Table 24-43.

Return to the Summary Table.

ADC Result 23 Register

Figure 24-67. ADCRESULT23 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-43. ADCRESULT23 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 23 16-bit ADC result. After the ADC completes a conversion of SOC23, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.25 ADCRESULT24 Register (Offset = 30h) [Reset = 0000h]

ADCRESULT24 is shown in Figure 24-68 and described in Table 24-44.

Return to the Summary Table.

ADC Result 24 Register

Figure 24-68. ADCRESULT24 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-44. ADCRESULT24 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 24 16-bit ADC result. After the ADC completes a conversion of SOC24, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.26 ADCRESULT25 Register (Offset = 32h) [Reset = 0000h]

ADCRESULT25 is shown in Figure 24-69 and described in Table 24-45.

Return to the Summary Table.

ADC Result 25 Register

Figure 24-69. ADCRESULT25 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-45. ADCRESULT25 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 25 16-bit ADC result. After the ADC completes a conversion of SOC25, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.27 ADCRESULT26 Register (Offset = 34h) [Reset = 0000h]

ADCRESULT26 is shown in Figure 24-70 and described in Table 24-46.

Return to the Summary Table.

ADC Result 26 Register

Figure 24-70. ADCRESULT26 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-46. ADCRESULT26 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 26 16-bit ADC result. After the ADC completes a conversion of SOC26, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.28 ADCRESULT27 Register (Offset = 36h) [Reset = 0000h]

ADCRESULT27 is shown in Figure 24-71 and described in Table 24-47.

Return to the Summary Table.

ADC Result 27 Register

Figure 24-71. ADCRESULT27 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-47. ADCRESULT27 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 27 16-bit ADC result. After the ADC completes a conversion of SOC27, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.29 ADCRESULT28 Register (Offset = 38h) [Reset = 0000h]

ADCRESULT28 is shown in Figure 24-72 and described in Table 24-48.

Return to the Summary Table.

ADC Result 28 Register

Figure 24-72. ADCRESULT28 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-48. ADCRESULT28 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 28 16-bit ADC result. After the ADC completes a conversion of SOC28, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.30 ADCRESULT29 Register (Offset = 3Ah) [Reset = 0000h]

ADCRESULT29 is shown in Figure 24-73 and described in Table 24-49.

Return to the Summary Table.

ADC Result 29 Register

Figure 24-73. ADCRESULT29 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-49. ADCRESULT29 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 29 16-bit ADC result. After the ADC completes a conversion of SOC29, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.31 ADCRESULT30 Register (Offset = 3Ch) [Reset = 0000h]

ADCRESULT30 is shown in Figure 24-74 and described in Table 24-50.

Return to the Summary Table.

ADC Result 30 Register

Figure 24-74. ADCRESULT30 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-50. ADCRESULT30 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 30 16-bit ADC result. After the ADC completes a conversion of SOC30, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.32 ADCRESULT31 Register (Offset = 3Eh) [Reset = 0000h]

ADCRESULT31 is shown in Figure 24-75 and described in Table 24-51.

Return to the Summary Table.

ADC Result 31 Register

Figure 24-75. ADCRESULT31 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 24-51. ADCRESULT31 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 31 16-bit ADC result. After the ADC completes a conversion of SOC31, the digital result is placed in this bit field. Reset type: SYSRSn

24.16.2.33 ADCPPB1RESULT Register (Offset = 40h) [Reset = 00000000h]

ADCPPB1RESULT is shown in Figure 24-76 and described in Table 24-52.

Return to the Summary Table.

ADC Post Processing Block 1 Result Register

Figure 24-76. ADCPPB1RESULT Register

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

Table 24-52. ADCPPB1RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the SIGN bits extend down to bit 12, and all reflect the same value as bit 12. Reset type: SYSRSn
15-0	PPBRESULT	R	0h	ADC Post Processing Block Result 1 The result of the offset/reference subtraction post conversion processing is stored in this register. This result is available 1 SYSCLK cycle after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 1 SYSCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles (refer to the TRM for more detailed timing information). If ADCINTFLG is polled to determine when to read the PPBRESULT, it may be necessary to add one or more NOP instructions to ensure that the updated post conversion processing result has posted to the register. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the PPBRESULT bits are limited to bits 12:0. Reset type: SYSRSn

24.16.2.34 ADCPPB2RESULT Register (Offset = 44h) [Reset = 00000000h]

ADCPPB2RESULT is shown in Figure 24-77 and described in Table 24-53.

Return to the Summary Table.

ADC Post Processing Block 2 Result Register

Figure 24-77. ADCPPB2RESULT Register

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

Table 24-53. ADCPPB2RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the SIGN bits extend down to bit 12, and all reflect the same value as bit 12. Reset type: SYSRSn
15-0	PPBRESULT	R	0h	ADC Post Processing Block Result 2 The result of the offset/reference subtraction post conversion processing is stored in this register. This result is available 1 SYSCLK cycle after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 1 SYSCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles (refer to the TRM for more detailed timing information). If ADCINTFLG is polled to determine when to read the PPBRESULT, it may be necessary to add one or more NOP instructions to ensure that the updated post conversion processing result has posted to the register. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the PPBRESULT bits are limited to bits 12:0. Reset type: SYSRSn

24.16.2.35 ADCPPB3RESULT Register (Offset = 48h) [Reset = 00000000h]

ADCPPB3RESULT is shown in Figure 24-78 and described in Table 24-54.

Return to the Summary Table.

ADC Post Processing Block 3 Result Register

Figure 24-78. ADCPPB3RESULT Register

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

Table 24-54. ADCPPB3RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the SIGN bits extend down to bit 12, and all reflect the same value as bit 12. Reset type: SYSRSn
15-0	PPBRESULT	R	0h	ADC Post Processing Block Result 3 The result of the offset/reference subtraction post conversion processing is stored in this register. This result is available 1 SYSCLK cycle after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 1 SYSCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles (refer to the TRM for more detailed timing information). If ADCINTFLG is polled to determine when to read the PPBRESULT, it may be necessary to add one or more NOP instructions to ensure that the updated post conversion processing result has posted to the register. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the PPBRESULT bits are limited to bits 12:0. Reset type: SYSRSn

24.16.2.36 ADCPPB4RESULT Register (Offset = 4Ch) [Reset = 00000000h]

ADCPPB4RESULT is shown in Figure 24-79 and described in Table 24-55.

Return to the Summary Table.

ADC Post Processing Block 4 Result Register

Figure 24-79. ADCPPB4RESULT Register

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

Table 24-55. ADCPPB4RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the SIGN bits extend down to bit 12, and all reflect the same value as bit 12. Reset type: SYSRSn
15-0	PPBRESULT	R	0h	ADC Post Processing Block Result 4 The result of the offset/reference subtraction post conversion processing is stored in this register. This result is available 1 SYSCLK cycle after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 1 SYSCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles (refer to the TRM for more detailed timing information). If ADCINTFLG is polled to determine when to read the PPBRESULT, it may be necessary to add one or more NOP instructions to ensure that the updated post conversion processing result has posted to the register. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the PPBRESULT bits are limited to bits 12:0. Reset type: SYSRSn

24.16.2.37 ADCPPB1SUM Register (Offset = 50h) [Reset = 00000000h]

ADCPPB1SUM is shown in Figure 24-80 and described in Table 24-56.

Return to the Summary Table.

ADC PPB 1 Final Sum Result Register

Figure 24-80. ADCPPB1SUM Register

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

Table 24-56. ADCPPB1SUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 23. Reset type: SYSRSn
23-0	SUM	R	0h	Post Processing Block 1 Oversampling Final Sum. When either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event, the value of PSUM is loaded into this register. In the case of a count-match event, the sum loaded into this register includes the value from the most recent conversion. The value from PSUM will be right shifted by the amount specified in the SHIFT register before being loaded into the final SUM result register. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

24.16.2.38 ADCPPB1COUNT Register (Offset = 54h) [Reset = 0000h]

ADCPPB1COUNT is shown in Figure 24-81 and described in Table 24-57.

Return to the Summary Table.

ADC PPB1 Final Conversion Count Register

Figure 24-81. ADCPPB1COUNT Register

15	14	13	12	11	10	9	8
RESERVED						COUNT	
R-0h						R-0h	
7	6	5	4	3	2	1	0
COUNT							
R-0h							

Table 24-57. ADCPPB1COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	COUNT	R	0h	Post Processing Block 1 Oversampling Final Count. When either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event, the value of PCOUNT is loaded into this register. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

24.16.2.39 ADCPPB2SUM Register (Offset = 58h) [Reset = 00000000h]

ADCPPB2SUM is shown in Figure 24-82 and described in Table 24-58.

Return to the Summary Table.

ADC PPB 2 Final Sum Result Register

Figure 24-82. ADCPPB2SUM Register

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

Table 24-58. ADCPPB2SUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 23. Reset type: SYSRSn
23-0	SUM	R	0h	Post Processing Block 2 Oversampling Final Sum. When either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event, the value of PSUM is loaded into this register. In the case of a count-match event, the sum loaded into this register includes the value from the most recent conversion. The value from PSUM will be right shifted by the amount specified in the SHIFT register before being loaded into the final SUM result register. This result is available 1 SYSCLK cycle after the associated ADCPPB2RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

24.16.2.40 ADCPPB2COUNT Register (Offset = 5Ch) [Reset = 0000h]

ADCPPB2COUNT is shown in Figure 24-83 and described in Table 24-59.

Return to the Summary Table.

ADC PPB2 Final Conversion Count Register

Figure 24-83. ADCPPB2COUNT Register

15	14	13	12	11	10	9	8
RESERVED						COUNT	
R-0h						R-0h	
7	6	5	4	3	2	1	0
COUNT							
R-0h							

Table 24-59. ADCPPB2COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	COUNT	R	0h	Post Processing Block 2 Oversampling Final Count. When either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event, the value of PCOUNT is loaded into this register. This result is available 1 SYSCLK cycle after the associated ADCPPB2RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

24.16.2.41 ADCPPB3SUM Register (Offset = 60h) [Reset = 00000000h]

ADCPPB3SUM is shown in Figure 24-84 and described in Table 24-60.

Return to the Summary Table.

ADC PPB 3 Final Sum Result Register

Figure 24-84. ADCPPB3SUM Register

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

Table 24-60. ADCPPB3SUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 23. Reset type: SYSRSn
23-0	SUM	R	0h	Post Processing Block 3 Oversampling Final Sum. When either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event, the value of PSUM is loaded into this register. In the case of a count-match event, the sum loaded into this register includes the value from the most recent conversion. The value from PSUM will be right shifted by the amount specified in the SHIFT register before being loaded into the final SUM result register. This result is available 1 SYSCLK cycle after the associated ADCPPB3RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

24.16.2.42 ADCPPB3COUNT Register (Offset = 64h) [Reset = 0000h]

ADCPPB3COUNT is shown in Figure 24-85 and described in Table 24-61.

Return to the Summary Table.

ADC PPB3 Final Conversion Count Register

Figure 24-85. ADCPPB3COUNT Register

15	14	13	12	11	10	9	8
RESERVED						COUNT	
R-0h						R-0h	
7	6	5	4	3	2	1	0
COUNT							
R-0h							

Table 24-61. ADCPPB3COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	COUNT	R	0h	Post Processing Block 3 Oversampling Final Count. When either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event, the value of PCOUNT is loaded into this register. This result is available 1 SYSCLK cycle after the associated ADCPPB3RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

24.16.2.43 ADCPPB4SUM Register (Offset = 68h) [Reset = 00000000h]

ADCPPB4SUM is shown in Figure 24-86 and described in Table 24-62.

Return to the Summary Table.

ADC PPB 4 Final Sum Result Register

Figure 24-86. ADCPPB4SUM Register

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

Table 24-62. ADCPPB4SUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 23. Reset type: SYSRSn
23-0	SUM	R	0h	Post Processing Block 4 Oversampling Final Sum. When either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event, the value of PSUM is loaded into this register. In the case of a count-match event, the sum loaded into this register includes the value from the most recent conversion. The value from PSUM will be right shifted by the amount specified in the SHIFT register before being loaded into the final SUM result register. This result is available 1 SYSCLK cycle after the associated ADCPPB4RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

24.16.2.44 ADCPPB4COUNT Register (Offset = 6Ch) [Reset = 0000h]

ADCPPB4COUNT is shown in Figure 24-87 and described in Table 24-63.

Return to the Summary Table.

ADC PPB4 Final Conversion Count Register

Figure 24-87. ADCPPB4COUNT Register

15	14	13	12	11	10	9	8
RESERVED						COUNT	
R-0h						R-0h	
7	6	5	4	3	2	1	0
COUNT							
R-0h							

Table 24-63. ADCPPB4COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	COUNT	R	0h	Post Processing Block 4 Oversampling Final Count. When either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event, the value of PCOUNT is loaded into this register. This result is available 1 SYSCLK cycle after the associated ADCPPB4RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

24.16.2.45 ADCPPB1MAX Register (Offset = 70h) [Reset = 00000000h]

ADCPPB1MAX is shown in Figure 24-88 and described in Table 24-64.

Return to the Summary Table.

ADC PPB 1 Final Max Result Register

Figure 24-88. ADCPPB1MAX Register

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

Table 24-64. ADCPPB1MAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. Reset type: SYSRSn
16-0	MAX	R	0h	Post Processing Block 1 Oversampling Final Max. When either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event, the value of PMAX is loaded into this register. In the case of a count-match event, the max loaded into this register includes the value from the most recent conversion. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available (only when a count-match event occurs). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

24.16.2.46 ADCPPB1MAXI Register (Offset = 74h) [Reset = 0000h]

ADCPPB1MAXI is shown in Figure 24-89 and described in Table 24-65.

Return to the Summary Table.

ADC PPB 1 Final Max Index Result Register

Figure 24-89. ADCPPB1MAXI Register

15	14	13	12	11	10	9	8
RESERVED						MAXI	
R-0h						R-0h	
7	6	5	4	3	2	1	0
MAXI							
R-0h							

Table 24-65. ADCPPB1MAXI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	MAXI	R	0h	Post Processing Block 1 Oversampling Final Index of the Max. When either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event, the value of PMAXI is loaded into this register. In the case of a count-match event, the max index loaded into this register includes the value from the most recent conversion. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

24.16.2.47 ADCPPB1MIN Register (Offset = 78h) [Reset = 00000000h]

ADCPPB1MIN is shown in Figure 24-90 and described in Table 24-66.

Return to the Summary Table.

ADC PPB 1 Final Min Result Register

Figure 24-90. ADCPPB1MIN Register

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

Table 24-66. ADCPPB1MIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. Reset type: SYSRSn
16-0	MIN	R	0h	Post Processing Block 1 Oversampling Final Min. When either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event, the value of PMIN is loaded into this register. In the case of a count-match event, the max loaded into this register includes the value from the most recent conversion. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

24.16.2.48 ADCPPB1MINI Register (Offset = 7Ch) [Reset = 0000h]

ADCPPB1MINI is shown in Figure 24-91 and described in Table 24-67.

Return to the Summary Table.

ADC PPB 1 Final Min Index Result Register

Figure 24-91. ADCPPB1MINI Register

15	14	13	12	11	10	9	8
RESERVED						MINI	
R-0h						R-0h	
7	6	5	4	3	2	1	0
MINI							
R-0h							

Table 24-67. ADCPPB1MINI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	MINI	R	0h	Post Processing Block 1 Oversampling Final Index of the Min. When either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event, the value of PMINI is loaded into this register. In the case of a count-match event, the min index loaded into this register includes the value from the most recent conversion. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

24.16.2.49 ADCPPB2MAX Register (Offset = 80h) [Reset = 00000000h]

ADCPPB2MAX is shown in Figure 24-92 and described in Table 24-68.

Return to the Summary Table.

ADC PPB 2 Final Max Result Register

Figure 24-92. ADCPPB2MAX Register

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

Table 24-68. ADCPPB2MAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. Reset type: SYSRSn
16-0	MAX	R	0h	Post Processing Block 2 Oversampling Final Max. When either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event, the value of PMAX is loaded into this register. In the case of a count-match event, the max loaded into this register includes the value from the most recent conversion. This result is available 1 SYSCLK cycle after the associated ADCPPB2RESULT is available (only when a count-match event occurs). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

24.16.2.50 ADCPPB2MAXI Register (Offset = 84h) [Reset = 0000h]

ADCPPB2MAXI is shown in Figure 24-93 and described in Table 24-69.

Return to the Summary Table.

ADC PPB 2 Final Max Index Result Register

Figure 24-93. ADCPPB2MAXI Register

15	14	13	12	11	10	9	8
RESERVED						MAXI	
R-0h						R-0h	
7	6	5	4	3	2	1	0
MAXI							
R-0h							

Table 24-69. ADCPPB2MAXI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	MAXI	R	0h	Post Processing Block 2 Oversampling Final Index of the Max. When either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event, the value of PMAXI is loaded into this register. In the case of a count-match event, the max index loaded into this register includes the value from the most recent conversion. This result is available 1 SYSCLK cycle after the associated ADCPPB2RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

24.16.2.51 ADCPPB2MIN Register (Offset = 88h) [Reset = 00000000h]

ADCPPB2MIN is shown in Figure 24-94 and described in Table 24-70.

Return to the Summary Table.

ADC PPB 2 Final Min Result Register

Figure 24-94. ADCPPB2MIN Register

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

Table 24-70. ADCPPB2MIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. Reset type: SYSRSn
16-0	MIN	R	0h	Post Processing Block 2 Oversampling Final Min. When either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event, the value of PMIN is loaded into this register. In the case of a count-match event, the max loaded into this register includes the value from the most recent conversion. This result is available 1 SYSCLK cycle after the associated ADCPPB2RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

24.16.2.52 ADCPPB2MINI Register (Offset = 8Ch) [Reset = 0000h]

ADCPPB2MINI is shown in Figure 24-95 and described in Table 24-71.

Return to the Summary Table.

ADC PPB 2 Final Min Index Result Register

Figure 24-95. ADCPPB2MINI Register

15	14	13	12	11	10	9	8
RESERVED						MINI	
R-0h						R-0h	
7	6	5	4	3	2	1	0
MINI							
R-0h							

Table 24-71. ADCPPB2MINI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	MINI	R	0h	Post Processing Block 2 Oversampling Final Index of the Min. When either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event, the value of PMINI is loaded into this register. In the case of a count-match event, the min index loaded into this register includes the value from the most recent conversion. This result is available 1 SYSCLK cycle after the associated ADCPPB2RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

24.16.2.53 ADCPPB3MAX Register (Offset = 90h) [Reset = 00000000h]

ADCPPB3MAX is shown in Figure 24-96 and described in Table 24-72.

Return to the Summary Table.

ADC PPB 3 Final Max Result Register

Figure 24-96. ADCPPB3MAX Register

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

Table 24-72. ADCPPB3MAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. Reset type: SYSRSn
16-0	MAX	R	0h	Post Processing Block 3 Oversampling Final Max. When either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event, the value of PMAX is loaded into this register. In the case of a count-match event, the max loaded into this register includes the value from the most recent conversion. This result is available 1 SYSCLK cycle after the associated ADCPPB3RESULT is available (only when a count-match event occurs). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

24.16.2.54 ADCPPB3MAXI Register (Offset = 94h) [Reset = 0000h]

ADCPPB3MAXI is shown in Figure 24-97 and described in Table 24-73.

Return to the Summary Table.

ADC PPB 3 Final Max Index Result Register

Figure 24-97. ADCPPB3MAXI Register

15	14	13	12	11	10	9	8
RESERVED						MAXI	
R-0h						R-0h	
7	6	5	4	3	2	1	0
MAXI							
R-0h							

Table 24-73. ADCPPB3MAXI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	MAXI	R	0h	Post Processing Block 3 Oversampling Final Index of the Max. When either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event, the value of PMAXI is loaded into this register. In the case of a count-match event, the max index loaded into this register includes the value from the most recent conversion. This result is available 1 SYSCLK cycle after the associated ADCPPB3RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

24.16.2.55 ADCPPB3MIN Register (Offset = 98h) [Reset = 00000000h]

ADCPPB3MIN is shown in Figure 24-98 and described in Table 24-74.

Return to the Summary Table.

ADC PPB 3 Final Min Result Register

Figure 24-98. ADCPPB3MIN Register

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

Table 24-74. ADCPPB3MIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. Reset type: SYSRSn
16-0	MIN	R	0h	Post Processing Block 3 Oversampling Final Min. When either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event, the value of PMIN is loaded into this register. In the case of a count-match event, the max loaded into this register includes the value from the most recent conversion. This result is available 1 SYSCLK cycle after the associated ADCPPB3RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

24.16.2.56 ADCPPB3MINI Register (Offset = 9Ch) [Reset = 0000h]

ADCPPB3MINI is shown in Figure 24-99 and described in Table 24-75.

Return to the Summary Table.

ADC PPB 3 Final Min Index Result Register

Figure 24-99. ADCPPB3MINI Register

15	14	13	12	11	10	9	8
RESERVED						MINI	
R-0h						R-0h	
7	6	5	4	3	2	1	0
MINI							
R-0h							

Table 24-75. ADCPPB3MINI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	MINI	R	0h	Post Processing Block 3 Oversampling Final Index of the Min. When either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event, the value of PMINI is loaded into this register. In the case of a count-match event, the min index loaded into this register includes the value from the most recent conversion. This result is available 1 SYSCLK cycle after the associated ADCPPB3RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

24.16.2.57 ADCPPB4MAX Register (Offset = A0h) [Reset = 00000000h]

ADCPPB4MAX is shown in Figure 24-100 and described in Table 24-76.

Return to the Summary Table.

ADC PPB 4 Final Max Result Register

Figure 24-100. ADCPPB4MAX Register

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

Table 24-76. ADCPPB4MAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. Reset type: SYSRSn
16-0	MAX	R	0h	Post Processing Block 4 Oversampling Final Max. When either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event, the value of PMAX is loaded into this register. In the case of a count-match event, the max loaded into this register includes the value from the most recent conversion. This result is available 1 SYSCLK cycle after the associated ADCPPB4RESULT is available (only when a count-match event occurs). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

24.16.2.58 ADCPPB4MAXI Register (Offset = A4h) [Reset = 0000h]

ADCPPB4MAXI is shown in Figure 24-101 and described in Table 24-77.

Return to the Summary Table.

ADC PPB 4 Final Max Index Result Register

Figure 24-101. ADCPPB4MAXI Register

15	14	13	12	11	10	9	8
RESERVED						MAXI	
R-0h						R-0h	
7	6	5	4	3	2	1	0
MAXI							
R-0h							

Table 24-77. ADCPPB4MAXI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	MAXI	R	0h	Post Processing Block 4 Oversampling Final Index of the Max. When either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event, the value of PMAXI is loaded into this register. In the case of a count-match event, the max index loaded into this register includes the value from the most recent conversion. This result is available 1 SYSCLK cycle after the associated ADCPPB4RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

24.16.2.59 ADCPPB4MIN Register (Offset = A8h) [Reset = 00000000h]

ADCPPB4MIN is shown in Figure 24-102 and described in Table 24-78.

Return to the Summary Table.

ADC PPB 4 Final Min Result Register

Figure 24-102. ADCPPB4MIN Register

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

Table 24-78. ADCPPB4MIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. Reset type: SYSRSn
16-0	MIN	R	0h	Post Processing Block 4 Oversampling Final Min. When either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event, the value of PMIN is loaded into this register. In the case of a count-match event, the max loaded into this register includes the value from the most recent conversion. This result is available 1 SYSCLK cycle after the associated ADCPPB4RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

24.16.2.60 ADCPPB4MINI Register (Offset = ACh) [Reset = 0000h]

ADCPPB4MINI is shown in Figure 24-103 and described in Table 24-79.

Return to the Summary Table.

ADC PPB 4 Final Min Index Result Register

Figure 24-103. ADCPPB4MINI Register

15	14	13	12	11	10	9	8
RESERVED						MINI	
R-0h						R-0h	
7	6	5	4	3	2	1	0
MINI							
R-0h							

Table 24-79. ADCPPB4MINI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	MINI	R	0h	Post Processing Block 4 Oversampling Final Index of the Min. When either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event, the value of PMINI is loaded into this register. In the case of a count-match event, the min index loaded into this register includes the value from the most recent conversion. This result is available 1 SYSCLK cycle after the associated ADCPPB4RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

24.16.3 ADC_REGS Registers

Table 24-80 lists the memory-mapped registers for the ADC_REGS registers. All register offset addresses not listed in Table 24-80 should be considered as reserved locations and the register contents should not be modified.

Table 24-80. ADC_REGS Registers

Offset	Acronym	Register Name	Protection
0h	ADCCTL1	ADC Control 1 Register	
2h	ADCCTL2	ADC Control 2 Register	
Ch	ADCBURSTCTL	ADC Burst Control Register	
Eh	ADCINTFLG	ADC Interrupt Flag Register	
10h	ADCINTFLGCLR	ADC Interrupt Flag Clear Register	
12h	ADCINTOVF	ADC Interrupt Overflow Register	
14h	ADCINTOVFCLR	ADC Interrupt Overflow Clear Register	
16h	ADCINTSEL1N2	ADC Interrupt 1 and 2 Selection Register	
18h	ADCINTSEL3N4	ADC Interrupt 3 and 4 Selection Register	
1Ah	ADCSOCPRICTL	ADC SOC Priority Control Register	
1Ch	ADCINTSOCSEL1	ADC Interrupt SOC Selection 1 Register	
20h	ADCINTSOCSEL2	ADC Interrupt SOC Selection 2 Register	
24h	ADCSOCFLG1	ADC SOC Flag 1 Register	
28h	ADCSOCFRC1	ADC SOC Force 1 Register	
2Ch	ADCSOCOVF1	ADC SOC Overflow 1 Register	
30h	ADCSOCOVFCLR1	ADC SOC Overflow Clear 1 Register	
34h	ADCSOC0CTL	ADC SOC0 Control Register	
38h	ADCSOC1CTL	ADC SOC1 Control Register	
3Ch	ADCSOC2CTL	ADC SOC2 Control Register	
40h	ADCSOC3CTL	ADC SOC3 Control Register	
44h	ADCSOC4CTL	ADC SOC4 Control Register	
48h	ADCSOC5CTL	ADC SOC5 Control Register	
4Ch	ADCSOC6CTL	ADC SOC6 Control Register	
50h	ADCSOC7CTL	ADC SOC7 Control Register	
54h	ADCSOC8CTL	ADC SOC8 Control Register	
58h	ADCSOC9CTL	ADC SOC9 Control Register	
5Ch	ADCSOC10CTL	ADC SOC10 Control Register	
60h	ADCSOC11CTL	ADC SOC11 Control Register	
64h	ADCSOC12CTL	ADC SOC12 Control Register	
68h	ADCSOC13CTL	ADC SOC13 Control Register	
6Ch	ADCSOC14CTL	ADC SOC14 Control Register	
70h	ADCSOC15CTL	ADC SOC15 Control Register	
74h	ADCSOC16CTL	ADC SOC16 Control Register	
78h	ADCSOC17CTL	ADC SOC17 Control Register	
7Ch	ADCSOC18CTL	ADC SOC18 Control Register	
80h	ADCSOC19CTL	ADC SOC19 Control Register	
84h	ADCSOC20CTL	ADC SOC20 Control Register	
88h	ADCSOC21CTL	ADC SOC21 Control Register	
8Ch	ADCSOC22CTL	ADC SOC22 Control Register	
90h	ADCSOC23CTL	ADC SOC23 Control Register	

Table 24-80. ADC_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
94h	ADCSOC24CTL	ADC SOC24 Control Register	
98h	ADCSOC25CTL	ADC SOC25 Control Register	
9Ch	ADCSOC26CTL	ADC SOC26 Control Register	
A0h	ADCSOC27CTL	ADC SOC27 Control Register	
A4h	ADCSOC28CTL	ADC SOC28 Control Register	
A8h	ADCSOC29CTL	ADC SOC29 Control Register	
ACh	ADCSOC30CTL	ADC SOC30 Control Register	
B0h	ADCSOC31CTL	ADC SOC31 Control Register	
B4h	ADCEVTSTAT	ADC Event Status Register	
B8h	ADCEVTCLR	ADC Event Clear Register	
BCh	ADCEVTSEL	ADC Event Selection Register	
C0h	ADCEVTINTSEL	ADC Event Interrupt Selection Register	
C4h	ADCOSDETECT	ADC Open and Shorts Detect Register	
C6h	ADCCOUNTER	ADC Counter Register	
C8h	ADCREV	ADC Revision Register	
CAh	ADCOFFTRIM	ADC Offset Trim Register 1	
CCh	ADCOFFTRIM2	ADC Offset Trim Register 2	
CEh	ADCOFFTRIM3	ADC Offset Trim Register 3	
D4h	ADCPPB1CONFIG	ADC PPB{#} Config Register	
D6h	ADCPPB1STAMP	ADC PPB1 Sample Delay Time Stamp Register	
D8h	ADCPPB1OFFCAL	ADC PPB1 Offset Calibration Register	
DAh	ADCPPB1OFFREF	ADC PPB1 Offset Reference Register	
DCh	ADCPPB1TRIPHI	ADC PPB1 Trip High Register	
E0h	ADCPPB1TRIPLO	ADC PPB1 Trip Low/Trigger Time Stamp Register	
E4h	ADCPPBTRIP1FILCTL	ADCEVT1 Trip High Filter Control Register	
E8h	ADCPPBTRIP1FILCLKCTL	ADCEVT1 Trip High Filter Prescale Control Register	
F4h	ADCPPB2CONFIG	ADC PPB{#} Config Register	
F6h	ADCPPB2STAMP	ADC PPB2 Sample Delay Time Stamp Register	
F8h	ADCPPB2OFFCAL	ADC PPB2 Offset Calibration Register	
FAh	ADCPPB2OFFREF	ADC PPB2 Offset Reference Register	
FCh	ADCPPB2TRIPHI	ADC PPB2 Trip High Register	
100h	ADCPPB2TRIPLO	ADC PPB2 Trip Low/Trigger Time Stamp Register	
104h	ADCPPBTRIP2FILCTL	ADCEVT2 Trip High Filter Control Register	
108h	ADCPPBTRIP2FILCLKCTL	ADCEVT2 Trip High Filter Prescale Control Register	
114h	ADCPPB3CONFIG	ADC PPB{#} Config Register	
116h	ADCPPB3STAMP	ADC PPB3 Sample Delay Time Stamp Register	
118h	ADCPPB3OFFCAL	ADC PPB3 Offset Calibration Register	
11Ah	ADCPPB3OFFREF	ADC PPB3 Offset Reference Register	
11Ch	ADCPPB3TRIPHI	ADC PPB3 Trip High Register	
120h	ADCPPB3TRIPLO	ADC PPB3 Trip Low/Trigger Time Stamp Register	
124h	ADCPPBTRIP3FILCTL	ADCEVT3 Trip High Filter Control Register	
128h	ADCPPBTRIP3FILCLKCTL	ADCEVT3 Trip High Filter Prescale Control Register	
134h	ADCPPB4CONFIG	ADC PPB{#} Config Register	
136h	ADCPPB4STAMP	ADC PPB4 Sample Delay Time Stamp Register	
138h	ADCPPB4OFFCAL	ADC PPB4 Offset Calibration Register	

Table 24-80. ADC_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
13Ah	ADCPPB4OFFREF	ADC PPB4 Offset Reference Register	
13Ch	ADCPPB4TRIPHI	ADC PPB4 Trip High Register	
140h	ADCPPB4TRIPLO	ADC PPB4 Trip Low/Trigger Time Stamp Register	
144h	ADCPPBTRIP4FILCTL	ADCEVT4 Trip High Filter Control Register	
148h	ADCPPBTRIP4FILCLKCTL	ADCEVT4 Trip High Filter Prescale Control Register	
154h	ADCSAFECHECKRESEN	ADC Safe Check Result Enable Register	
158h	ADCSAFECHECKRESEN2	ADC Safe Check Result Enable 2 Register	
172h	ADCINTCYCLE	ADC Early Interrupt Generation Cycle	
174h	ADCINLTRIM1	ADC Linearity Trim 1 Register	
178h	ADCINLTRIM2	ADC Linearity Trim 2 Register	
17Ch	ADCINLTRIM3	ADC Linearity Trim 3 Register	
180h	ADCINLTRIM4	ADC Linearity Trim 4 Register	
184h	ADCINLTRIM5	ADC Linearity Trim 5 Register	
188h	ADCINLTRIM6	ADC Linearity Trim 6 Register	
18Eh	ADCREV2	ADC Wrapper Revision Register	
194h	REP1CTL	ADC Trigger Repeater 1 Control Register	
198h	REP1N	ADC Trigger Repeater 1 N Select Register	
19Ch	REP1PHASE	ADC Trigger Repeater 1 Phase Select Register	
1A0h	REP1SPREAD	ADC Trigger Repeater 1 Spread Select Register	
1A4h	REP1FRC	ADC Trigger Repeater 1 Software Force Register	
1B4h	REP2CTL	ADC Trigger Repeater 2 Control Register	
1B8h	REP2N	ADC Trigger Repeater 2 N Select Register	
1BCh	REP2PHASE	ADC Trigger Repeater 2 Phase Select Register	
1C0h	REP2SPREAD	ADC Trigger Repeater 2 Spread Select Register	
1C4h	REP2FRC	ADC Trigger Repeater 2 Software Force Register	
1D4h	ADCPPB1LIMIT	ADC PPB1Conversion Count Limit Register	
1D8h	ADCPPBP1PCOUNT	ADC PPB1 Partial Conversion Count Register	
1DCh	ADCPPB1CONFIG2	ADC PPB1 Sum Shift Register	
1E0h	ADCPPB1PSUM	ADC PPB1 Partial Sum Register	
1E4h	ADCPPB1PMAX	ADC PPB1 Partial Max Register	
1E8h	ADCPPB1PMAXI	ADC PPB1 Partial Max Index Register	
1ECh	ADCPPB1PMIN	ADC PPB1 Partial MIN Register	
1F0h	ADCPPB1PMINI	ADC PPB1 Partial Min Index Register	
1F4h	ADCPPB1TRIPLO2	ADC PPB1 Extended Trip Low Register	
208h	ADCPPB2LIMIT	ADC PPB2Conversion Count Limit Register	
20Ch	ADCPPBP2PCOUNT	ADC PPB2 Partial Conversion Count Register	
210h	ADCPPB2CONFIG2	ADC PPB2 Sum Shift Register	
214h	ADCPPB2PSUM	ADC PPB2 Partial Sum Register	
218h	ADCPPB2PMAX	ADC PPB2 Partial Max Register	
21Ch	ADCPPB2PMAXI	ADC PPB2 Partial Max Index Register	
220h	ADCPPB2PMIN	ADC PPB2 Partial MIN Register	
224h	ADCPPB2PMINI	ADC PPB2 Partial Min Index Register	
228h	ADCPPB2TRIPLO2	ADC PPB2 Extended Trip Low Register	
23Ch	ADCPPB3LIMIT	ADC PPB3Conversion Count Limit Register	
240h	ADCPPBP3PCOUNT	ADC PPB3 Partial Conversion Count Register	

Table 24-80. ADC_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
244h	ADCPPB3CONFIG2	ADC PPB3 Sum Shift Register	
248h	ADCPPB3PSUM	ADC PPB3 Partial Sum Register	
24Ch	ADCPPB3PMAX	ADC PPB3 Partial Max Register	
250h	ADCPPB3PMAXI	ADC PPB3 Partial Max Index Register	
254h	ADCPPB3PMIN	ADC PPB3 Partial MIN Register	
258h	ADCPPB3PMINI	ADC PPB3 Partial Min Index Register	
25Ch	ADCPPB3TRIPLO2	ADC PPB3 Extended Trip Low Register	
270h	ADCPPB4LIMIT	ADC PPB4Conversion Count Limit Register	
274h	ADCPPBP4PCOUNT	ADC PPB4 Partial Conversion Count Register	
278h	ADCPPB4CONFIG2	ADC PPB4 Sum Shift Register	
27Ch	ADCPPB4PSUM	ADC PPB4 Partial Sum Register	
280h	ADCPPB4PMAX	ADC PPB4 Partial Max Register	
284h	ADCPPB4PMAXI	ADC PPB4 Partial Max Index Register	
288h	ADCPPB4PMIN	ADC PPB4 Partial MIN Register	
28Ch	ADCPPB4PMINI	ADC PPB4 Partial Min Index Register	
290h	ADCPPB4TRIPLO2	ADC PPB4 Extended Trip Low Register	

Complex bit access types are encoded to fit into small table cells. Table 24-81 shows the codes that are used for access types in this section.

Table 24-81. ADC_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

24.16.3.1 ADCCTL1 Register (Offset = 0h) [Reset = 0000h]

ADCCTL1 is shown in Figure 24-104 and described in Table 24-82.

Return to the Summary Table.

ADC Control 1 Register

Figure 24-104. ADCCTL1 Register

15	14	13	12	11	10	9	8
TDMAEN	EXTMUXPRESELECTEN	ADCBSY	ADCBSYCHN				
R/W-0h	R/W-0h	R-0h	R-0h				
7	6	5	4	3	2	1	0
ADCPWDNZ	RESERVED				INTPULSEPOS	RESERVED	
R/W-0h	R-0h				R/W-0h	R-0h	

Table 24-82. ADCCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	TDMAEN	R/W	0h	ADC Power Down (active low). This bit controls the power up and power down of all the analog circuitry inside the analog core. 0 DMA is triggered at the same time as the CPU interrupt 1 DMA is always triggered at tDMA regardless or whether the ADC is in early interrupt mode or late interrupt mode Reset type: SYSRSn
14	EXTMUXPRESELECTEN	R/W	0h	If th the ADC SOC sequence is deterministic, the ADCEXTMUX pins can be set earlier: at the end of the S+H window of the previous conversion instead of the beginning of the S+H window of the current conversion. This allows some of the external mux settling time to be pipelined with the previous conversion's conversion time. However, this will not work in the case where high-priority SOCs can arrive asynchronously. 0 ADCEXTMUX pins only change at beginning of S+H window 1 ADCEXTMUX pins are set after the end of S+H window based on pending SOCs Reset type: SYSRSn
13	ADCBSY	R	0h	ADC Busy. Set when ADC SOC is generated, cleared by hardware four ADC clocks after negative edge of S/H pulse. Used by the ADC state machine to determine if ADC is available to sample. 0 ADC is available to sample next channel 1 ADC is busy and cannot sample another channel Reset type: SYSRSn

Table 24-82. ADCCTL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12-8	ADCBSYCHN	R	0h	ADC Busy Channel. Set when an ADC Start of Conversion (SOC) is generated. When ADCBSY=0: holds the value of the last converted SOC When ADCBSY=1: reflects the SOC currently being processed 00h SOC0 is currently processing or was last SOC converted 01h SOC1 is currently processing or was last SOC converted 02h SOC2 is currently processing or was last SOC converted 03h SOC3 is currently processing or was last SOC converted 04h SOC4 is currently processing or was last SOC converted 05h SOC5 is currently processing or was last SOC converted 06h SOC6 is currently processing or was last SOC converted 07h SOC7 is currently processing or was last SOC converted 08h SOC8 is currently processing or was last SOC converted 09h SOC9 is currently processing or was last SOC converted 0Ah SOC10 is currently processing or was last SOC converted 0Bh SOC11 is currently processing or was last SOC converted 0Ch SOC12 is currently processing or was last SOC converted 0Dh SOC13 is currently processing or was last SOC converted 0Eh SOC14 is currently processing or was last SOC converted 0Fh SOC15 is currently processing or was last SOC converted 10h SOC16 is currently processing or was last SOC converted 11h SOC17 is currently processing or was last SOC converted 12h SOC18 is currently processing or was last SOC converted 13h SOC19 is currently processing or was last SOC converted 14h SOC20 is currently processing or was last SOC converted 15h SOC21 is currently processing or was last SOC converted 16h SOC22 is currently processing or was last SOC converted 17h SOC23 is currently processing or was last SOC converted 18h SOC24 is currently processing or was last SOC converted 19h SOC25 is currently processing or was last SOC converted 1Ah SOC26 is currently processing or was last SOC converted 1Bh SOC27 is currently processing or was last SOC converted 1Ch SOC28 is currently processing or was last SOC converted 1Dh SOC29 is currently processing or was last SOC converted 1Eh SOC30 is currently processing or was last SOC converted 1Fh SOC31 is currently processing or was last SOC converted Reset type: SYSRSn
7	ADCPWDNZ	R/W	0h	ADC Power Down (active low). This bit controls the power up and power down of all the analog circuitry inside the analog core. 0 All analog circuitry inside the core is powered down 1 All analog circuitry inside the core is powered up Reset type: SYSRSn
6-3	RESERVED	R	0h	Reserved
2	INTPULSEPOS	R/W	0h	ADC Interrupt Pulse Position. 0 Interrupt pulse generation occurs when ADC begins conversion (at the end of the acquisition window) plus a number of SYSCLK cycles as specified in the ADCINTCYCLE.OFFSET register. 1 Interrupt pulse generation occurs at the end of the conversion, 1 cycle prior to the ADC result latching into its result register Reset type: SYSRSn
1-0	RESERVED	R	0h	Reserved

24.16.3.2 ADCCTL2 Register (Offset = 2h) [Reset = 0000h]

ADCCTL2 is shown in Figure 24-105 and described in Table 24-83.

Return to the Summary Table.

ADC Control 2 Register

Figure 24-105. ADCCTL2 Register

15	14	13	12	11	10	9	8
RESERVED			RESERVED				OFFTRIMMODE
R-0h			R-0h				R/W-0h
7	6	5	4	3	2	1	0
SIGNALMODE	RESOLUTION	RESERVED		PRESCALE			
R/W-0h	R/W-0h	R-0h		R/W-0h			

Table 24-83. ADCCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12-9	RESERVED	R	0h	Reserved
8	OFFTRIMMODE	R/W	0h	ADC offset trim mode. 0 = Offset trim supplied by ADCOFFTRIM.OFFTRIM regardless of resolution or signal mode 1 = Offset trim for each combination of resolution, signalmode, and even or odd is supplied by a different field in ADCOFFTRIM, ADCOFFTRIM2, or ADCOFFTRIM3 Reset type: SYSRSn
7	SIGNALMODE	R/W	0h	SOC Signaling Mode. Selects the input mode of the converter. Use the AdcSetMode function to change the signal mode. 0 Single-ended 1 Differential Reset type: SYSRSn
6	RESOLUTION	R/W	0h	SOC Conversion Resolution. Selects the resolution of the converter. Use the AdcSetMode function to change the resolution. 0 12-bit resolution 1 16-bit resolution Reset type: SYSRSn
5-4	RESERVED	R	0h	Reserved
3-0	PRESCALE	R/W	0h	ADC Clock Prescaler. 0000 ADCCLK = Input Clock / 1.0 0001 Invalid 0010 ADCCLK = Input Clock / 2.0 0011 ADCCLK = Input Clock / 2.5 0100 ADCCLK = Input Clock / 3.0 0101 ADCCLK = Input Clock / 3.5 0110 ADCCLK = Input Clock / 4.0 0111 ADCCLK = Input Clock / 4.5 1000 ADCCLK = Input Clock / 5.0 1001 ADCCLK = Input Clock / 5.5 1010 ADCCLK = Input Clock / 6.0 1011 ADCCLK = Input Clock / 6.5 1100 ADCCLK = Input Clock / 7.0 1101 ADCCLK = Input Clock / 7.5 1110 ADCCLK = Input Clock / 8.0 1111 ADCCLK = Input Clock / 8.5 Reset type: SYSRSn

24.16.3.3 ADCBURSTCTL Register (Offset = Ch) [Reset = 0000h]

ADCBURSTCTL is shown in Figure 24-106 and described in Table 24-84.

Return to the Summary Table.

ADC Burst Control Register

Figure 24-106. ADCBURSTCTL Register

15	14	13	12	11	10	9	8
BURSTEN	RESERVED		BURSTSIZE				
R/W-0h	R-0h		R/W-0h				
7	6	5	4	3	2	1	0
RESERVED	BURSTTRIGSEL						
R-0h	R/W-0h						

Table 24-84. ADCBURSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	BURSTEN	R/W	0h	SOC Burst Mode Enable. This bit enables the SOC Burst Mode of operation. 0 Burst mode is disabled. 1 Burst mode is enabled. Reset type: SYSRSn
14-13	RESERVED	R	0h	Reserved

Table 24-84. ADCBURSTCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12-8	BURSTSIZE	R/W	0h	SOC Burst Size Select. This bit field determines how many SOCs are converted when a burst conversion sequence is started. The first SOC converted is defined by the round robin pointer, which is advanced as each SOC is converted. 0h 1 SOC converted 1h 2 SOCs converted 2h 3 SOCs converted 3h 4 SOCs converted 4h 5 SOCs converted 5h 6 SOCs converted 6h 7 SOCs converted 7h 8 SOCs converted 8h 9 SOCs converted 9h 10 SOCs converted Ah 11 SOCs converted Bh 12 SOCs converted Ch 13 SOCs converted Dh 14 SOCs converted Eh 15 SOCs converted Fh 16 SOCs converted 10h 17 SOC converted 11h 18 SOCs converted 12h 19 SOCs converted 13h 20 SOCs converted 14h 21 SOCs converted 15h 22 SOCs converted 16h 23 SOCs converted 17h 24 SOCs converted 18h 25 SOCs converted 19h 26 SOCs converted 1Ah 27 SOCs converted 1Bh 28 SOCs converted 1Ch 29 SOCs converted 1Dh 30 SOCs converted 1Eh 31 SOCs converted 1Fh 32 SOCs converted Note: If the burst causes SOCs to be set for conversion that were already pending, the corresponding bits in the ADCSOCOVF register will be set. Reset type: SYSRSn
7	RESERVED	R	0h	Reserved

Table 24-84. ADCBURSTCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6-0	BURSTTRIGSEL	R/W	0h	SOC Burst Trigger Source Select. Configures which trigger will start a burst conversion sequence. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h BURSTTRIG0 - Software only 01h BURSTTRIG1 - CPU1 Timer 0, TINT0n 02h BURSTTRIG2 - CPU1 Timer 1, TINT1n 03h BURSTTRIG3 - CPU1 Timer 2, TINT2n 04h BURSTTRIG4 - GPIO, Input X-Bar INPUT5 05h BURSTTRIG5 - ePWM1, ADCSOCA 06h BURSTTRIG6 - ePWM1, ADCSOCB 07h BURSTTRIG7 - ePWM2, ADCSOCA 08h BURSTTRIG8 - ePWM2, ADCSOCB 09h BURSTTRIG9 - ePWM3, ADCSOCA 0Ah BURSTTRIG10 - ePWM3, ADCSOCB 0Bh BURSTTRIG11 - ePWM4, ADCSOCA 0Ch BURSTTRIG12 - ePWM4, ADCSOCB 0Dh BURSTTRIG13 - ePWM5, ADCSOCA 0Eh BURSTTRIG14 - ePWM5, ADCSOCB 0Fh BURSTTRIG15 - ePWM6, ADCSOCA 10h BURSTTRIG16 - ePWM6, ADCSOCB 11h BURSTTRIG17 - ePWM7, ADCSOCA 12h BURSTTRIG18 - ePWM7, ADCSOCB 13h BURSTTRIG19 - ePWM8, ADCSOCA 14h BURSTTRIG20 - ePWM8, ADCSOCB 15h BURSTTRIG21 - ePWM9, ADCSOCA 16h BURSTTRIG22 - ePWM9, ADCSOCB 17h BURSTTRIG23 - ePWM10, ADCSOCA 18h BURSTTRIG24 - ePWM10, ADCSOCB 19h BURSTTRIG25 - ePWM11, ADCSOCA 1Ah BURSTTRIG26 - ePWM11, ADCSOCB 1Bh BURSTTRIG27 - ePWM12, ADCSOCA 1Ch BURSTTRIG28 - ePWM12, ADCSOCB 1Dh - 1Fh - Reserved 20h BURSTTRIG32 - ePWM13, ADCSOCA 21h BURSTTRIG33 - ePWM13, ADCSOCB 22h BURSTTRIG34 - ePWM14, ADCSOCA 23h BURSTTRIG35 - ePWM14, ADCSOCB 24h BURSTTRIG36 - ePWM15, ADCSOCA 25h BURSTTRIG37 - ePWM15, ADCSOCB 26h BURSTTRIG38 - ePWM16, ADCSOCA 27h BURSTTRIG39 - ePWM16, ADCSOCB 28h BURSTTRIG40 - REP1TRIG 29h BURSTTRIG41 - REP2TRIG 2Ah - 2Fh - Reserved 30h BURSTTRIG48 - ePWM17, ADCSOCA 31h BURSTTRIG49 - ePWM17, ADCSOCB 32h BURSTTRIG50 - ePWM18, ADCSOCA 33h BURSTTRIG51 - ePWM18, ADCSOCB 34h BURSTTRIG52 - ePWM19, ADCSOCA 35h BURSTTRIG53 - ePWM19, ADCSOCB 36h BURSTTRIG54 - ePWM20, ADCSOCA 37h BURSTTRIG55 - ePWM20, ADCSOCB 38h BURSTTRIG56 - ePWM21, ADCSOCA 39h BURSTTRIG57 - ePWM21, ADCSOCB 3Ah BURSTTRIG58 - ePWM22, ADCSOCA 3Bh BURSTTRIG59 - ePWM22, ADCSOCB 3Ch BURSTTRIG60 - ePWM23, ADCSOCA 3Dh BURSTTRIG61 - ePWM23, ADCSOCB 3Eh BURSTTRIG62 - ePWM24, ADCSOCA 3Fh BURSTTRIG63 - ePWM24, ADCSOCB 40h BURSTTRIG64 - ePWM25, ADCSOCA 41h BURSTTRIG65 - ePWM25, ADCSOCB 42h BURSTTRIG66 - ePWM26, ADCSOCA 43h BURSTTRIG67 - ePWM26, ADCSOCB

Table 24-84. ADCBURSTCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
				44h BURSTTRIG68 - ePWM27, ADCSOCA 45h BURSTTRIG69 - ePWM27, ADCSOCB 46h BURSTTRIG70 - ePWM28, ADCSOCA 47h BURSTTRIG71 - ePWM28, ADCSOCB 48h BURSTTRIG72 - ePWM29, ADCSOCA 49h BURSTTRIG73 - ePWM29, ADCSOCB 4Ah BURSTTRIG74 - ePWM30, ADCSOCA 4Bh BURSTTRIG75 - ePWM30, ADCSOCB 4Ch BURSTTRIG76 - ePWM31, ADCSOCA 4Dh BURSTTRIG77 - ePWM31, ADCSOCB 4Eh BURSTTRIG78 - ePWM32, ADCSOCA 4Fh BURSTTRIG79 - ePWM32, ADCSOCB 50h BURSTTRIG80 eCAP1 51h BURSTTRIG81 eCAP2 52h BURSTTRIG82 eCAP3 53h BURSTTRIG83 eCAP4 54h BURSTTRIG84 eCAP5 55h BURSTTRIG85 eCAP6 56h BURSTTRIG86 eCAP7 57h BURSTTRIG87 eCAP8 58h - 7Fh - Reserved Reset type: SYSRSn

24.16.3.4 ADCINTFLG Register (Offset = Eh) [Reset = 0000h]

ADCINTFLG is shown in Figure 24-107 and described in Table 24-85.

Return to the Summary Table.

ADC Interrupt Flag Register

Figure 24-107. ADCINTFLG Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
ADCINT4RESULT	ADCINT3RESULT	ADCINT2RESULT	ADCINT1RESULT	ADCINT4	ADCINT3	ADCINT2	ADCINT1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 24-85. ADCINTFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	ADCINT4RESULT	R	0h	ADC Interrupt 4 Results Ready Flag. This flag is set when the conversions results associated with ADCINT4 latch into the corresponding results register. 0 Conversion results have not latched 1 Conversion results have latched This flag can be used in an ISR that is entered in early interrupt mode to ensure that the corresponding results are ready before proceeding to read the result register. This flag can be cleared via the ACK bit in the ADCINTFLGCLR that also clears the ADCINT4 flag. In case results latch and this flag is already set, the corresponding flag in ADCINTOVF is NOT set. This flag does NOT have to be cleared in order for ADCINT ISRs to propagate to the PIE. In case the associated SOC is associated with a PPB or PPBs, the flag will not be set until all associated PPB results latch. Reset type: SYSRSn
6	ADCINT3RESULT	R	0h	ADC Interrupt 3 Results Ready Flag. This flag is set when the conversions results associated with ADCINT3 latch into the corresponding results register. 0 Conversion results have not latched 1 Conversion results have latched This flag can be used in an ISR that is entered in early interrupt mode to ensure that the corresponding results are ready before proceeding to read the result register. This flag can be cleared via the ACK bit in the ADCINTFLGCLR that also clears the ADCINT3 flag. In case results latch and this flag is already set, the corresponding flag in ADCINTOVF is NOT set. This flag does NOT have to be cleared in order for ADCINT ISRs to propagate to the PIE. In case the associated SOC is associated with a PPB or PPBs, the flag will not be set until all associated PPB results latch. Reset type: SYSRSn

Table 24-85. ADCINTFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	ADCINT2RESULT	R	0h	ADC Interrupt 2 Results Ready Flag. This flag is set when the conversions results associated with ADCINT2 latch into the corresponding results register. 0 Conversion results have not latched 1 Conversion results have latched This flag can be used in an ISR that is entered in early interrupt mode to ensure that the corresponding results are ready before proceeding to read the result register. This flag can be cleared via the ACK bit in the ADCINTFLGCLR that also clears the ADCINT2 flag. In case results latch and this flag is already set, the corresponding flag in ADCINTOVF is NOT set. This flag does NOT have to be cleared in order for ADCINT ISRs to propagate to the PIE. In case the associated SOC is associated with a PPB or PPBs, the flag will not be set until all associated PPB results latch. Reset type: SYSRSn
4	ADCINT1RESULT	R	0h	ADC Interrupt 1 Results Ready Flag. This flag is set when the conversions results associated with ADCINT1 latch into the corresponding results register. 0 Conversion results have not latched 1 Conversion results have latched This flag can be used in an ISR that is entered in early interrupt mode to ensure that the corresponding results are ready before proceeding to read the result register. This flag can be cleared via the ACK bit in the ADCINTFLGCLR that also clears the ADCINT1 flag. In case results latch and this flag is already set, the corresponding flag in ADCINTOVF is NOT set. This flag does NOT have to be cleared in order for ADCINT ISRs to propagate to the PIE. In case the associated SOC is associated with a PPB or PPBs, the flag will not be set until all associated PPB results latch. Reset type: SYSRSn
3	ADCINT4	R	0h	ADC Interrupt 4 Flag. Reading these flags indicates if the associated ADCINT pulse was generated since the last clear. 0 No ADC interrupt pulse generated 1 ADC interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (INTSELxNy register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn
2	ADCINT3	R	0h	ADC Interrupt 3 Flag. Reading these flags indicates if the associated ADCINT pulse was generated since the last clear. 0 No ADC interrupt pulse generated 1 ADC interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (INTSELxNy register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn

Table 24-85. ADCINTFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	ADCINT2	R	0h	ADC Interrupt 2 Flag. Reading these flags indicates if the associated ADCINT pulse was generated since the last clear. 0 No ADC interrupt pulse generated 1 ADC interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (INTSELxNy register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn
0	ADCINT1	R	0h	ADC Interrupt 1 Flag. Reading these flags indicates if the associated ADCINT pulse was generated since the last clear. 0 No ADC interrupt pulse generated 1 ADC interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (INTSELxNy register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn

24.16.3.5 ADCINTFLGCLR Register (Offset = 10h) [Reset = 0000h]

ADCINTFLGCLR is shown in Figure 24-108 and described in Table 24-86.

Return to the Summary Table.

ADC Interrupt Flag Clear Register

Figure 24-108. ADCINTFLGCLR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				ADCINT4	ADCINT3	ADCINT2	ADCINT1
R-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 24-86. ADCINTFLGCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3	ADCINT4	R-0/W1C	0h	ADC Interrupt 4 Flag Clear. Reads return 0. 0 No action 1 Clears ADCINT4 and ADCINT4RESULT flags in the ADCINTFLG register. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
2	ADCINT3	R-0/W1C	0h	ADC Interrupt 3 Flag Clear. Reads return 0. 0 No action 1 Clears ADCINT3 and ADCINT3RESULT flags in the ADCINTFLG register. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
1	ADCINT2	R-0/W1C	0h	ADC Interrupt 2 Flag Clear. Reads return 0. 0 No action 1 Clears ADCINT2 and ADCINT2RESULT flags in the ADCINTFLG register. . If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
0	ADCINT1	R-0/W1C	0h	ADC Interrupt 1 Flag Clear. Reads return 0. 0 No action 1 Clears ADCINT1 and ADCINT1RESULT flags in the ADCINTFLG register. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn

24.16.3.6 ADCINTOVF Register (Offset = 12h) [Reset = 0000h]

ADCINTOVF is shown in Figure 24-109 and described in Table 24-87.

Return to the Summary Table.

ADC Interrupt Overflow Register

Figure 24-109. ADCINTOVF Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				ADCINT4	ADCINT3	ADCINT2	ADCINT1
R-0h				R-0h	R-0h	R-0h	R-0h

Table 24-87. ADCINTOVF Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3	ADCINT4	R	0h	ADC Interrupt 4 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC interrupt overflow event detected. 1 ADC Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
2	ADCINT3	R	0h	ADC Interrupt 3 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC interrupt overflow event detected. 1 ADC Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
1	ADCINT2	R	0h	ADC Interrupt 2 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC interrupt overflow event detected. 1 ADC Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
0	ADCINT1	R	0h	ADC Interrupt 1 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC interrupt overflow event detected. 1 ADC Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn

24.16.3.7 ADCINTOVFCLR Register (Offset = 14h) [Reset = 0000h]

ADCINTOVFCLR is shown in Figure 24-110 and described in Table 24-88.

Return to the Summary Table.

ADC Interrupt Overflow Clear Register

Figure 24-110. ADCINTOVFCLR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				ADCINT4	ADCINT3	ADCINT2	ADCINT1
R-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 24-88. ADCINTOVFCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3	ADCINT4	R-0/W1C	0h	ADC Interrupt 4 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
2	ADCINT3	R-0/W1C	0h	ADC Interrupt 3 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
1	ADCINT2	R-0/W1C	0h	ADC Interrupt 2 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
0	ADCINT1	R-0/W1C	0h	ADC Interrupt 1 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn

24.16.3.8 ADCINTSEL1N2 Register (Offset = 16h) [Reset = 0000h]

ADCINTSEL1N2 is shown in Figure 24-111 and described in Table 24-89.

Return to the Summary Table.

ADC Interrupt 1 and 2 Selection Register

Figure 24-111. ADCINTSEL1N2 Register

15	14	13	12	11	10	9	8
INT2E	INT2CONT	INT2SEL					
R/W-0h	R/W-0h	R/W-0h					
7	6	5	4	3	2	1	0
INT1E	INT1CONT	INT1SEL					
R/W-0h	R/W-0h	R/W-0h					

Table 24-89. ADCINTSEL1N2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	INT2E	R/W	0h	ADCINT2 Interrupt Enable 0 ADCINT2 is disabled 1 ADCINT2 is enabled Reset type: SYSRSn
14	INT2CONT	R/W	0h	ADCINT2 Continue to Interrupt Mode 0 No further ADCINT2 pulses are generated until ADCINT2 flag (in ADCINTFLG register) is cleared by user. 1 ADCINT2 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn

Table 24-89. ADCINTSEL1N2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-8	INT2SEL	R/W	0h	ADCINT2 EOC Source Select 00h EOC0 is trigger for ADCINT2 01h EOC1 is trigger for ADCINT2 02h EOC2 is trigger for ADCINT2 03h EOC3 is trigger for ADCINT2 04h EOC4 is trigger for ADCINT2 05h EOC5 is trigger for ADCINT2 06h EOC6 is trigger for ADCINT2 07h EOC7 is trigger for ADCINT2 08h EOC8 is trigger for ADCINT2 09h EOC9 is trigger for ADCINT2 0Ah EOC10 is trigger for ADCINT2 0Bh EOC11 is trigger for ADCINT2 0Ch EOC12 is trigger for ADCINT2 0Dh EOC13 is trigger for ADCINT2 0Eh EOC14 is trigger for ADCINT2 0Fh EOC15 is trigger for ADCINT2 10h EOC16 is trigger for ADCINT2 11h EOC17 is trigger for ADCINT2 12h EOC18 is trigger for ADCINT2 13h EOC19 is trigger for ADCINT2 14h EOC20 is trigger for ADCINT2 15h EOC21 is trigger for ADCINT2 16h EOC22 is trigger for ADCINT2 17h EOC23 is trigger for ADCINT2 18h EOC24 is trigger for ADCINT2 19h EOC25 is trigger for ADCINT2 1Ah EOC26 is trigger for ADCINT2 1Bh EOC27 is trigger for ADCINT2 1Ch EOC28 is trigger for ADCINT2 1Dh EOC29 is trigger for ADCINT2 1Eh EOC30 is trigger for ADCINT2 1Fh EOC31 is trigger for ADCINT2 20h OSINT1 is trigger for ADCINT2 21h OSINT2 is trigger for ADCINT2 22h OSINT3 is trigger for ADCINT2 23h OSINT4 is trigger for ADCINT2 Reset type: SYSRSn
7	INT1E	R/W	0h	ADCINT1 Interrupt Enable 0 ADCINT1 is disabled 1 ADCINT1 is enabled Reset type: SYSRSn
6	INT1CONT	R/W	0h	ADCINT1 Continue to Interrupt Mode 0 No further ADCINT1 pulses are generated until ADCINT1 flag (in ADCINTFLG register) is cleared by user. 1 ADCINT1 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn

Table 24-89. ADCINTSEL1N2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-0	INT1SEL	R/W	0h	ADCINT1 EOC Source Select 00h EOC0 is trigger for ADCINT1 01h EOC1 is trigger for ADCINT1 02h EOC2 is trigger for ADCINT1 03h EOC3 is trigger for ADCINT1 04h EOC4 is trigger for ADCINT1 05h EOC5 is trigger for ADCINT1 06h EOC6 is trigger for ADCINT1 07h EOC7 is trigger for ADCINT1 08h EOC8 is trigger for ADCINT1 09h EOC9 is trigger for ADCINT1 0Ah EOC10 is trigger for ADCINT1 0Bh EOC11 is trigger for ADCINT1 0Ch EOC12 is trigger for ADCINT1 0Dh EOC13 is trigger for ADCINT1 0Eh EOC14 is trigger for ADCINT1 0Fh EOC15 is trigger for ADCINT1 10h EOC16 is trigger for ADCINT1 11h EOC17 is trigger for ADCINT1 12h EOC18 is trigger for ADCINT1 13h EOC19 is trigger for ADCINT1 14h EOC20 is trigger for ADCINT1 15h EOC21 is trigger for ADCINT1 16h EOC22 is trigger for ADCINT1 17h EOC23 is trigger for ADCINT1 18h EOC24 is trigger for ADCINT1 19h EOC25 is trigger for ADCINT1 1Ah EOC26 is trigger for ADCINT1 1Bh EOC27 is trigger for ADCINT1 1Ch EOC28 is trigger for ADCINT1 1Dh EOC29 is trigger for ADCINT1 1Eh EOC30 is trigger for ADCINT1 1Fh EOC31 is trigger for ADCINT1 20h OSINT1 is trigger for ADCINT1 21h OSINT2 is trigger for ADCINT1 22h OSINT3 is trigger for ADCINT1 23h OSINT4 is trigger for ADCINT1 Reset type: SYSRSn

24.16.3.9 ADCINTSEL3N4 Register (Offset = 18h) [Reset = 0000h]

ADCINTSEL3N4 is shown in Figure 24-112 and described in Table 24-90.

Return to the Summary Table.

ADC Interrupt 3 and 4 Selection Register

Figure 24-112. ADCINTSEL3N4 Register

15	14	13	12	11	10	9	8
INT4E	INT4CONT	INT4SEL					
R/W-0h	R/W-0h	R/W-0h					
7	6	5	4	3	2	1	0
INT3E	INT3CONT	INT3SEL					
R/W-0h	R/W-0h	R/W-0h					

Table 24-90. ADCINTSEL3N4 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	INT4E	R/W	0h	ADCINT4 Interrupt Enable 0 ADCINT4 is disabled 1 ADCINT4 is enabled Reset type: SYSRSn
14	INT4CONT	R/W	0h	ADCINT4 Continue to Interrupt Mode 0 No further ADCINT4 pulses are generated until ADCINT4 flag (in ADCINTFLG register) is cleared by user. 1 ADCINT4 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn

Table 24-90. ADCINTSEL3N4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-8	INT4SEL	R/W	0h	ADCINT4 EOC Source Select 00h EOC0 is trigger for ADCINT4 01h EOC1 is trigger for ADCINT4 02h EOC2 is trigger for ADCINT4 03h EOC3 is trigger for ADCINT4 04h EOC4 is trigger for ADCINT4 05h EOC5 is trigger for ADCINT4 06h EOC6 is trigger for ADCINT4 07h EOC7 is trigger for ADCINT4 08h EOC8 is trigger for ADCINT4 09h EOC9 is trigger for ADCINT4 0Ah EOC10 is trigger for ADCINT4 0Bh EOC11 is trigger for ADCINT4 0Ch EOC12 is trigger for ADCINT4 0Dh EOC13 is trigger for ADCINT4 0Eh EOC14 is trigger for ADCINT4 0Fh EOC15 is trigger for ADCINT4 10h EOC16 is trigger for ADCINT4 11h EOC17 is trigger for ADCINT4 12h EOC18 is trigger for ADCINT4 13h EOC19 is trigger for ADCINT4 14h EOC20 is trigger for ADCINT4 15h EOC21 is trigger for ADCINT4 16h EOC22 is trigger for ADCINT4 17h EOC23 is trigger for ADCINT4 18h EOC24 is trigger for ADCINT4 19h EOC25 is trigger for ADCINT4 1Ah EOC26 is trigger for ADCINT4 1Bh EOC27 is trigger for ADCINT4 1Ch EOC28 is trigger for ADCINT4 1Dh EOC29 is trigger for ADCINT4 1Eh EOC30 is trigger for ADCINT4 1Fh EOC31 is trigger for ADCINT4 20h OSINT1 is trigger for ADCINT4 21h OSINT2 is trigger for ADCINT4 22h OSINT3 is trigger for ADCINT4 23h OSINT4 is trigger for ADCINT4 Reset type: SYSRSn
7	INT3E	R/W	0h	ADCINT3 Interrupt Enable 0 ADCINT3 is disabled 1 ADCINT3 is enabled Reset type: SYSRSn
6	INT3CONT	R/W	0h	ADCINT3 Continue to Interrupt Mode 0 No further ADCINT3 pulses are generated until ADCINT3 flag (in ADCINTFLG register) is cleared by user. 1 ADCINT3 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn

Table 24-90. ADCINTSEL3N4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-0	INT3SEL	R/W	0h	ADCINT3 EOC Source Select 00h EOC0 is trigger for ADCINT3 01h EOC1 is trigger for ADCINT3 02h EOC2 is trigger for ADCINT3 03h EOC3 is trigger for ADCINT3 04h EOC4 is trigger for ADCINT3 05h EOC5 is trigger for ADCINT3 06h EOC6 is trigger for ADCINT3 07h EOC7 is trigger for ADCINT3 08h EOC8 is trigger for ADCINT3 09h EOC9 is trigger for ADCINT3 0Ah EOC10 is trigger for ADCINT3 0Bh EOC11 is trigger for ADCINT3 0Ch EOC12 is trigger for ADCINT3 0Dh EOC13 is trigger for ADCINT3 0Eh EOC14 is trigger for ADCINT3 0Fh EOC15 is trigger for ADCINT3 10h EOC16 is trigger for ADCINT3 11h EOC17 is trigger for ADCINT3 12h EOC18 is trigger for ADCINT3 13h EOC19 is trigger for ADCINT3 14h EOC20 is trigger for ADCINT3 15h EOC21 is trigger for ADCINT3 16h EOC22 is trigger for ADCINT3 17h EOC23 is trigger for ADCINT3 18h EOC24 is trigger for ADCINT3 19h EOC25 is trigger for ADCINT3 1Ah EOC26 is trigger for ADCINT3 1Bh EOC27 is trigger for ADCINT3 1Ch EOC28 is trigger for ADCINT3 1Dh EOC29 is trigger for ADCINT3 1Eh EOC30 is trigger for ADCINT3 1Fh EOC31 is trigger for ADCINT3 20h OSINT1 is trigger for ADCINT3 21h OSINT2 is trigger for ADCINT3 22h OSINT3 is trigger for ADCINT3 23h OSINT4 is trigger for ADCINT3 Reset type: SYSRSn

24.16.3.10 ADCSOCPRICTL Register (Offset = 1Ah) [Reset = 0800h]

ADCSOCPRICTL is shown in Figure 24-113 and described in Table 24-91.

Return to the Summary Table.

ADC SOC Priority Control Register

Figure 24-113. ADCSOCPRICTL Register

15	14	13	12	11	10	9	8
RESERVED				RRPOINTER			
R-0h				R-20h			
7	6	5	4	3	2	1	0
RRPOINTER		SOCPRIORITY					
R-20h		R/W-0h					

Table 24-91. ADCSOCPRICTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved

Table 24-91. ADCSOCPRICTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-6	RRPOINTER	R	20h	Round Robin Pointer. Holds the value of the last converted round robin SOCx to be used by the round robin scheme to determine order of conversions. 00h SOC0 was last round robin SOC to convert, SOC1 is highest round robin priority. 01h SOC1 was last round robin SOC to convert, SOC2 is highest round robin priority. 02h SOC2 was last round robin SOC to convert, SOC3 is highest round robin priority. 03h SOC3 was last round robin SOC to convert, SOC4 is highest round robin priority. 04h SOC4 was last round robin SOC to convert, SOC5 is highest round robin priority. 05h SOC5 was last round robin SOC to convert, SOC6 is highest round robin priority. 06h SOC6 was last round robin SOC to convert, SOC7 is highest round robin priority. 07h SOC7 was last round robin SOC to convert, SOC8 is highest round robin priority. 08h SOC8 was last round robin SOC to convert, SOC9 is highest round robin priority. 09h SOC9 was last round robin SOC to convert, SOC10 is highest round robin priority. 0Ah SOC10 was last round robin SOC to convert, SOC11 is highest round robin priority. 0Bh SOC11 was last round robin SOC to convert, SOC12 is highest round robin priority. 0Ch SOC12 was last round robin SOC to convert, SOC13 is highest round robin priority. 0Dh SOC13 was last round robin SOC to convert, SOC14 is highest round robin priority. 0Eh SOC14 was last round robin SOC to convert, SOC15 is highest round robin priority. 0Fh SOC15 was last round robin SOC to convert, SOC16 is highest round robin priority. 10h SOC16 was last round robin SOC to convert, SOC17 is highest round robin priority. 11h SOC17 was last round robin SOC to convert, SOC18 is highest round robin priority. 12h SOC18 was last round robin SOC to convert, SOC19 is highest round robin priority. 13h SOC19 was last round robin SOC to convert, SOC20 is highest round robin priority. 14h SOC20 was last round robin SOC to convert, SOC21 is highest round robin priority. 15h SOC21 was last round robin SOC to convert, SOC22 is highest round robin priority. 16h SOC22 was last round robin SOC to convert, SOC23 is highest round robin priority. 17h SOC23 was last round robin SOC to convert, SOC24 is highest round robin priority. 18h SOC24 was last round robin SOC to convert, SOC25 is highest round robin priority. 19h SOC25 was last round robin SOC to convert, SOC26 is highest round robin priority. 1Ah SOC26 was last round robin SOC to convert, SOC27 is highest round robin priority. 1Bh SOC27 was last round robin SOC to convert, SOC28 is highest round robin priority. 1Ch SOC28 was last round robin SOC to convert, SOC29 is highest round robin priority. 1Dh SOC29 was last round robin SOC to convert, SOC30 is highest round robin priority. 1Eh SOC30 was last round robin SOC to convert, SOC31 is highest round robin priority.

Table 24-91. ADCSOCPRICTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
				1Fh SOC31 was last round robin SOC to convert, SOC0 is highest round robin priority. 20h Reset value to indicate no SOC has been converted. SOC0 is highest round robin priority. Set to this value when the ADC module is reset by SOFTPRES or when the ADCSOCPRICTL register is written. In the latter case, if a conversion is currently in progress, it will complete and then the new priority will take effect. Others Invalid value. Reset type: SYSRSn

Table 24-91. ADCSOCPRICTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-0	SOCPRIORITY	R/W	0h	SOC Priority Determines the cutoff point for priority mode and round robin arbitration for SOCx 00h SOC priority is handled in round robin mode for all channels. 01h SOC0 is high priority, rest of channels are in round robin mode. 02h SOC0-SOC1 are high priority, SOC2 and greater are in round robin mode. 03h SOC0-SOC2 are high priority, SOC3 and greater are in round robin mode. 04h SOC0-SOC3 are high priority, SOC4 and greater are in round robin mode. 05h SOC0-SOC4 are high priority, SOC5 and greater are in round robin mode. 06h SOC0-SOC5 are high priority, SOC6 and greater are in round robin mode. 07h SOC0-SOC6 are high priority, SOC7 and greater are in round robin mode. 08h SOC0-SOC7 are high priority, SOC8 and greater are in round robin mode. 09h SOC0-SOC8 are high priority, SOC9 and greater are in round robin mode. 0Ah SOC0-SOC9 are high priority, SOC10 and greater are in round robin mode. 0Bh SOC0-SOC10 are high priority, SOC11 and greater are in round robin mode. 0Ch SOC0-SOC11 are high priority, SOC12 and greater are in round robin mode. 0Dh SOC0-SOC12 are high priority, SOC13 and greater are in round robin mode. 0Eh SOC0-SOC13 are high priority, SOC14 and greater are in round robin mode. 0Fh SOC0-SOC14 are high priority, SOC15 and greater are in round robin mode. 10h SOC0-SOC15 are high priority, SOC16 and greater are in round robin mode. 11h SOC0-SOC16 are high priority, SOC17 and greater are in round robin mode. 12h SOC0-SOC17 are high priority, SOC18 and greater are in round robin mode. 13h SOC0-SOC18 are high priority, SOC19 and greater are in round robin mode. 14h SOC0-SOC19 are high priority, SOC20 and greater are in round robin mode. 15h SOC0-SOC20 are high priority, SOC21 and greater are in round robin mode. 16h SOC0-SOC21 are high priority, SOC22 and greater are in round robin mode. 17h SOC0-SOC22 are high priority, SOC23 and greater are in round robin mode. 18h SOC0-SOC23 are high priority, SOC24 and greater are in round robin mode. 19h SOC0-SOC24 are high priority, SOC25 and greater are in round robin mode. 1Ah SOC0-SOC25 are high priority, SOC26 and greater are in round robin mode. 1Bh SOC0-SOC26 are high priority, SOC27 and greater are in round robin mode. 1Ch SOC0-SOC27 are high priority, SOC28 and greater are in round robin mode. 1Dh SOC0-SOC28 are high priority, SOC29 and greater are in round robin mode. 1Eh SOC0-SOC29 are high priority, SOC30 and greater are in round robin mode. 1Fh SOC0-SOC30 are high priority, SOC31 is in round robin mode. 20h All SOCs are in high priority mode, arbitrated by SOC number.

Table 24-91. ADCSOCPRICTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
				Others Invalid selection. Reset type: SYSRSn

24.16.3.11 ADCINTSOCSEL1 Register (Offset = 1Ch) [Reset = 00000000h]

ADCINTSOCSEL1 is shown in Figure 24-114 and described in Table 24-92.

Return to the Summary Table.

ADC Interrupt SOC Selection 1 Register

Figure 24-114. ADCINTSOCSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SOC15		SOC14		SOC13		SOC12		SOC11		SOC10		SOC9		SOC8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOC7		SOC6		SOC5		SOC4		SOC3		SOC2		SOC1		SOC0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 24-92. ADCINTSOCSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SOC15	R/W	0h	SOC15 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC15. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC15. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC15. 10 ADCINT2 will trigger SOC15. 11 Invalid selection. Reset type: SYSRSn
29-28	SOC14	R/W	0h	SOC14 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC14. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC14. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC14. 10 ADCINT2 will trigger SOC14. 11 Invalid selection. Reset type: SYSRSn
27-26	SOC13	R/W	0h	SOC13 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC13. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC13. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC13. 10 ADCINT2 will trigger SOC13. 11 Invalid selection. Reset type: SYSRSn
25-24	SOC12	R/W	0h	SOC12 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC12. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC12. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC12. 10 ADCINT2 will trigger SOC12. 11 Invalid selection. Reset type: SYSRSn

Table 24-92. ADCINTSOCSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-22	SOC11	R/W	0h	SOC11 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC11. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC11. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC11. 10 ADCINT2 will trigger SOC11. 11 Invalid selection. Reset type: SYSRSn
21-20	SOC10	R/W	0h	SOC10 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC10. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC10. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC10. 10 ADCINT2 will trigger SOC10. 11 Invalid selection. Reset type: SYSRSn
19-18	SOC9	R/W	0h	SOC9 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC9. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC9. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC9. 10 ADCINT2 will trigger SOC9. 11 Invalid selection. Reset type: SYSRSn
17-16	SOC8	R/W	0h	SOC8 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC8. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC8. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC8. 10 ADCINT2 will trigger SOC8. 11 Invalid selection. Reset type: SYSRSn
15-14	SOC7	R/W	0h	SOC7 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC7. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC7. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC7. 10 ADCINT2 will trigger SOC7. 11 Invalid selection. Reset type: SYSRSn
13-12	SOC6	R/W	0h	SOC6 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC6. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC6. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC6. 10 ADCINT2 will trigger SOC6. 11 Invalid selection. Reset type: SYSRSn

Table 24-92. ADCINTSOCSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-10	SOC5	R/W	0h	SOC5 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC5. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC5. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC5. 10 ADCINT2 will trigger SOC5. 11 Invalid selection. Reset type: SYSRSn
9-8	SOC4	R/W	0h	SOC4 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC4. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC4. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC4. 10 ADCINT2 will trigger SOC4. 11 Invalid selection. Reset type: SYSRSn
7-6	SOC3	R/W	0h	SOC3 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC3. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC3. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC3. 10 ADCINT2 will trigger SOC3. 11 Invalid selection. Reset type: SYSRSn
5-4	SOC2	R/W	0h	SOC2 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC2. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC2. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC2. 10 ADCINT2 will trigger SOC2. 11 Invalid selection. Reset type: SYSRSn
3-2	SOC1	R/W	0h	SOC1 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC1. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC1. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC1. 10 ADCINT2 will trigger SOC1. 11 Invalid selection. Reset type: SYSRSn
1-0	SOC0	R/W	0h	SOC0 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC0. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC0. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC0. 10 ADCINT2 will trigger SOC0. 11 Invalid selection. Reset type: SYSRSn

24.16.3.12 ADCINTSOCSEL2 Register (Offset = 20h) [Reset = 00000000h]

ADCINTSOCSEL2 is shown in Figure 24-115 and described in Table 24-93.

Return to the Summary Table.

ADC Interrupt SOC Selection 2 Register

Figure 24-115. ADCINTSOCSEL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SOC31		SOC30		SOC29		SOC28		SOC27		SOC26		SOC25		SOC24	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOC23		SOC22		SOC21		SOC20		SOC19		SOC18		SOC17		SOC16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 24-93. ADCINTSOCSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SOC31	R/W	0h	SOC31 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC31. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC31. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC31. 10 ADCINT2 will trigger SOC31. 11 Invalid selection. Reset type: SYSRSn
29-28	SOC30	R/W	0h	SOC30 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC30. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC30. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC30. 10 ADCINT2 will trigger SOC30. 11 Invalid selection. Reset type: SYSRSn
27-26	SOC29	R/W	0h	SOC29 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC29. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC29. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC29. 10 ADCINT2 will trigger SOC29. 11 Invalid selection. Reset type: SYSRSn
25-24	SOC28	R/W	0h	SOC28 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC28. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC28. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC28. 10 ADCINT2 will trigger SOC28. 11 Invalid selection. Reset type: SYSRSn

Table 24-93. ADCINTSOCSEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-22	SOC27	R/W	0h	SOC27 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC27. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC27. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC27. 10 ADCINT2 will trigger SOC27. 11 Invalid selection. Reset type: SYSRSn
21-20	SOC26	R/W	0h	SOC26 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC26. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC26. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC26. 10 ADCINT2 will trigger SOC26. 11 Invalid selection. Reset type: SYSRSn
19-18	SOC25	R/W	0h	SOC25 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC25. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC25. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC25. 10 ADCINT2 will trigger SOC25. 11 Invalid selection. Reset type: SYSRSn
17-16	SOC24	R/W	0h	SOC24 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC24. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC24. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC24. 10 ADCINT2 will trigger SOC24. 11 Invalid selection. Reset type: SYSRSn
15-14	SOC23	R/W	0h	SOC23 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC23. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC23. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC23. 10 ADCINT2 will trigger SOC23. 11 Invalid selection. Reset type: SYSRSn
13-12	SOC22	R/W	0h	SOC22 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC22. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC22. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC22. 10 ADCINT2 will trigger SOC22. 11 Invalid selection. Reset type: SYSRSn

Table 24-93. ADCINTSOCSEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-10	SOC21	R/W	0h	SOC21 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC21. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC21. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC21. 10 ADCINT2 will trigger SOC21. 11 Invalid selection. Reset type: SYSRSn
9-8	SOC20	R/W	0h	SOC20 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC20. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC20. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC20. 10 ADCINT2 will trigger SOC20. 11 Invalid selection. Reset type: SYSRSn
7-6	SOC19	R/W	0h	SOC19 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC19. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC19. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC19. 10 ADCINT2 will trigger SOC19. 11 Invalid selection. Reset type: SYSRSn
5-4	SOC18	R/W	0h	SOC18 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC18. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC18. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC18. 10 ADCINT2 will trigger SOC18. 11 Invalid selection. Reset type: SYSRSn
3-2	SOC17	R/W	0h	SOC17 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC17. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC17. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC17. 10 ADCINT2 will trigger SOC17. 11 Invalid selection. Reset type: SYSRSn
1-0	SOC16	R/W	0h	SOC16 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC16. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC16. TRIGSEL field alone determines SOC16 trigger. 01 ADCINT1 will trigger SOC16. 10 ADCINT2 will trigger SOC16. 11 Invalid selection. Reset type: SYSRSn

24.16.3.13 ADCSOCFLG1 Register (Offset = 24h) [Reset = 00000000h]

ADCSOCFLG1 is shown in Figure 24-116 and described in Table 24-94.

Return to the Summary Table.

ADC SOC Flag 1 Register

Figure 24-116. ADCSOCFLG1 Register

31	30	29	28	27	26	25	24
SOC31	SOC30	SOC29	SOC28	SOC27	SOC26	SOC25	SOC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
SOC23	SOC22	SOC21	SOC20	SOC19	SOC18	SOC17	SOC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
SOC7	SOC6	SOC5	SOC4	SOC3	SOC2	SOC1	SOC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 24-94. ADCSOCFLG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SOC31	R	0h	SOC31 Start of Conversion Flag. Indicates the state of SOC31 conversions. 0 No sample pending for SOC15. 1 Trigger has been received and sample is pending for SOC15. This bit will be automatically cleared when the SOC15 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
30	SOC30	R	0h	SOC30 Start of Conversion Flag. Indicates the state of SOC30 conversions. 0 No sample pending for SOC14. 1 Trigger has been received and sample is pending for SOC14. This bit will be automatically cleared when the SOC14 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

Table 24-94. ADCSOCFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
29	SOC29	R	0h	SOC29 Start of Conversion Flag. Indicates the state of SOC29 conversions. 0 No sample pending for SOC13. 1 Trigger has been received and sample is pending for SOC13. This bit will be automatically cleared when the SOC13 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
28	SOC28	R	0h	SOC28 Start of Conversion Flag. Indicates the state of SOC28 conversions. 0 No sample pending for SOC12. 1 Trigger has been received and sample is pending for SOC12. This bit will be automatically cleared when the SOC12 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
27	SOC27	R	0h	SOC27 Start of Conversion Flag. Indicates the state of SOC27 conversions. 0 No sample pending for SOC11. 1 Trigger has been received and sample is pending for SOC11. This bit will be automatically cleared when the SOC11 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
26	SOC26	R	0h	SOC26 Start of Conversion Flag. Indicates the state of SOC26 conversions. 0 No sample pending for SOC10. 1 Trigger has been received and sample is pending for SOC10. This bit will be automatically cleared when the SOC10 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
25	SOC25	R	0h	SOC25 Start of Conversion Flag. Indicates the state of SOC25 conversions. 0 No sample pending for SOC9. 1 Trigger has been received and sample is pending for SOC9. This bit will be automatically cleared when the SOC9 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

Table 24-94. ADCSOCFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	SOC24	R	0h	SOC24 Start of Conversion Flag. Indicates the state of SOC24 conversions. 0 No sample pending for SOC8. 1 Trigger has been received and sample is pending for SOC8. This bit will be automatically cleared when the SOC8 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
23	SOC23	R	0h	SOC23 Start of Conversion Flag. Indicates the state of SO23 conversions. 0 No sample pending for SOC7. 1 Trigger has been received and sample is pending for SOC7. This bit will be automatically cleared when the SOC7 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
22	SOC22	R	0h	SOC22 Start of Conversion Flag. Indicates the state of SOC22 conversions. 0 No sample pending for SOC6. 1 Trigger has been received and sample is pending for SOC6. This bit will be automatically cleared when the SOC6 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
21	SOC21	R	0h	SOC21 Start of Conversion Flag. Indicates the state of SOC21 conversions. 0 No sample pending for SOC5. 1 Trigger has been received and sample is pending for SOC5. This bit will be automatically cleared when the SOC5 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
20	SOC20	R	0h	SOC20 Start of Conversion Flag. Indicates the state of SOC20 conversions. 0 No sample pending for SOC4. 1 Trigger has been received and sample is pending for SOC4. This bit will be automatically cleared when the SOC4 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

Table 24-94. ADCSOCFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	SOC19	R	0h	SOC19 Start of Conversion Flag. Indicates the state of SOC19 conversions. 0 No sample pending for SOC3. 1 Trigger has been received and sample is pending for SOC3. This bit will be automatically cleared when the SOC3 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
18	SOC18	R	0h	SOC18 Start of Conversion Flag. Indicates the state of SOC18 conversions. 0 No sample pending for SOC2. 1 Trigger has been received and sample is pending for SOC2. This bit will be automatically cleared when the SOC2 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
17	SOC17	R	0h	SOC17 Start of Conversion Flag. Indicates the state of SOC17 conversions. 0 No sample pending for SOC1. 1 Trigger has been received and sample is pending for SOC1. This bit will be automatically cleared when the SOC1 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
16	SOC16	R	0h	SOC16 Start of Conversion Flag. Indicates the state of SO16 conversions. 0 No sample pending for SOC0. 1 Trigger has been received and sample is pending for SOC0. This bit will be automatically cleared when the SOC0 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
15	SOC15	R	0h	SOC15 Start of Conversion Flag. Indicates the state of SOC15 conversions. 0 No sample pending for SOC15. 1 Trigger has been received and sample is pending for SOC15. This bit will be automatically cleared when the SOC15 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

Table 24-94. ADCSOCFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	SOC14	R	0h	SOC14 Start of Conversion Flag. Indicates the state of SOC14 conversions. 0 No sample pending for SOC14. 1 Trigger has been received and sample is pending for SOC14. This bit will be automatically cleared when the SOC14 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
13	SOC13	R	0h	SOC13 Start of Conversion Flag. Indicates the state of SOC13 conversions. 0 No sample pending for SOC13. 1 Trigger has been received and sample is pending for SOC13. This bit will be automatically cleared when the SOC13 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
12	SOC12	R	0h	SOC12 Start of Conversion Flag. Indicates the state of SOC12 conversions. 0 No sample pending for SOC12. 1 Trigger has been received and sample is pending for SOC12. This bit will be automatically cleared when the SOC12 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
11	SOC11	R	0h	SOC11 Start of Conversion Flag. Indicates the state of SOC11 conversions. 0 No sample pending for SOC11. 1 Trigger has been received and sample is pending for SOC11. This bit will be automatically cleared when the SOC11 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
10	SOC10	R	0h	SOC10 Start of Conversion Flag. Indicates the state of SOC10 conversions. 0 No sample pending for SOC10. 1 Trigger has been received and sample is pending for SOC10. This bit will be automatically cleared when the SOC10 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

Table 24-94. ADCSOCFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	SOC9	R	0h	SOC9 Start of Conversion Flag. Indicates the state of SOC9 conversions. 0 No sample pending for SOC9. 1 Trigger has been received and sample is pending for SOC9. This bit will be automatically cleared when the SOC9 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
8	SOC8	R	0h	SOC8 Start of Conversion Flag. Indicates the state of SOC8 conversions. 0 No sample pending for SOC8. 1 Trigger has been received and sample is pending for SOC8. This bit will be automatically cleared when the SOC8 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
7	SOC7	R	0h	SOC7 Start of Conversion Flag. Indicates the state of SOC7 conversions. 0 No sample pending for SOC7. 1 Trigger has been received and sample is pending for SOC7. This bit will be automatically cleared when the SOC7 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
6	SOC6	R	0h	SOC6 Start of Conversion Flag. Indicates the state of SOC6 conversions. 0 No sample pending for SOC6. 1 Trigger has been received and sample is pending for SOC6. This bit will be automatically cleared when the SOC6 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
5	SOC5	R	0h	SOC5 Start of Conversion Flag. Indicates the state of SOC5 conversions. 0 No sample pending for SOC5. 1 Trigger has been received and sample is pending for SOC5. This bit will be automatically cleared when the SOC5 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

Table 24-94. ADCSOCFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	SOC4	R	0h	SOC4 Start of Conversion Flag. Indicates the state of SOC4 conversions. 0 No sample pending for SOC4. 1 Trigger has been received and sample is pending for SOC4. This bit will be automatically cleared when the SOC4 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
3	SOC3	R	0h	SOC3 Start of Conversion Flag. Indicates the state of SOC3 conversions. 0 No sample pending for SOC3. 1 Trigger has been received and sample is pending for SOC3. This bit will be automatically cleared when the SOC3 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
2	SOC2	R	0h	SOC2 Start of Conversion Flag. Indicates the state of SOC2 conversions. 0 No sample pending for SOC2. 1 Trigger has been received and sample is pending for SOC2. This bit will be automatically cleared when the SOC2 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
1	SOC1	R	0h	SOC1 Start of Conversion Flag. Indicates the state of SOC1 conversions. 0 No sample pending for SOC1. 1 Trigger has been received and sample is pending for SOC1. This bit will be automatically cleared when the SOC1 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
0	SOC0	R	0h	SOC0 Start of Conversion Flag. Indicates the state of SOC0 conversions. 0 No sample pending for SOC0. 1 Trigger has been received and sample is pending for SOC0. This bit will be automatically cleared when the SOC0 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

24.16.3.14 ADCSOCFRC1 Register (Offset = 28h) [Reset = 00000000h]

ADCSOCFRC1 is shown in Figure 24-117 and described in Table 24-95.

Return to the Summary Table.

ADC SOC Force 1 Register

Figure 24-117. ADCSOCFRC1 Register

31	30	29	28	27	26	25	24
SOC31	SOC30	SOC29	SOC28	SOC27	SOC26	SOC25	SOC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
SOC23	SOC22	SOC21	SOC20	SOC19	SOC18	SOC17	SOC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
SOC7	SOC6	SOC5	SOC4	SOC3	SOC2	SOC1	SOC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 24-95. ADCSOCFRC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SOC31	R-0/W1S	0h	SOC31 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC31 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC31 flag bit to 1. This will cause a conversion to start once priority is given to SOC31. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC31 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
30	SOC30	R-0/W1S	0h	SOC30 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC30 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC30 flag bit to 1. This will cause a conversion to start once priority is given to SOC30. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC30 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn

Table 24-95. ADCSOCFRC1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
29	SOC29	R-0/W1S	0h	SOC29 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC29 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC29 flag bit to 1. This will cause a conversion to start once priority is given to SOC29. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC29 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
28	SOC28	R-0/W1S	0h	SOC28 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC28 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC28 flag bit to 1. This will cause a conversion to start once priority is given to SOC28. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC28 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
27	SOC27	R-0/W1S	0h	SOC27 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC27 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC27 flag bit to 1. This will cause a conversion to start once priority is given to SOC27. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC27 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
26	SOC26	R-0/W1S	0h	SOC26 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC26 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC26 flag bit to 1. This will cause a conversion to start once priority is given to SOC26. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC26 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn

Table 24-95. ADCSOCFRC1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	SOC25	R-0/W1S	0h	SOC25 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC25 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC25 flag bit to 1. This will cause a conversion to start once priority is given to SOC25. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC25 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
24	SOC24	R-0/W1S	0h	SOC24 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC24 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC24 flag bit to 1. This will cause a conversion to start once priority is given to SOC24. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC24 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
23	SOC23	R-0/W1S	0h	SOC23 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC23 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC23 flag bit to 1. This will cause a conversion to start once priority is given to SOC23. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC23 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
22	SOC22	R-0/W1S	0h	SOC22 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC22 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC22 flag bit to 1. This will cause a conversion to start once priority is given to SOC22. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC22 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn

Table 24-95. ADCSOCFRC1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	SOC21	R-0/W1S	0h	SOC21 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC21 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC21 flag bit to 1. This will cause a conversion to start once priority is given to SOC21. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC21 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
20	SOC20	R-0/W1S	0h	SOC20 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC20 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC20 flag bit to 1. This will cause a conversion to start once priority is given to SOC20. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC20 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
19	SOC19	R-0/W1S	0h	SOC19 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC19 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC19 flag bit to 1. This will cause a conversion to start once priority is given to SOC19. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC19 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
18	SOC18	R-0/W1S	0h	SOC18 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC18 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC18 flag bit to 1. This will cause a conversion to start once priority is given to SOC18. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC18 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn

Table 24-95. ADCSOCFRC1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	SOC17	R-0/W1S	0h	SOC17 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC17 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC17 flag bit to 1. This will cause a conversion to start once priority is given to SOC17. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC17 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
16	SOC16	R-0/W1S	0h	SOC16 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC16 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC16 flag bit to 1. This will cause a conversion to start once priority is given to SOC16. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC16 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
15	SOC15	R-0/W1S	0h	SOC15 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC15 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC15 flag bit to 1. This will cause a conversion to start once priority is given to SOC15. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC15 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
14	SOC14	R-0/W1S	0h	SOC14 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC14 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC14 flag bit to 1. This will cause a conversion to start once priority is given to SOC14. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC14 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn

Table 24-95. ADCSOCFRC1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	SOC13	R-0/W1S	0h	SOC13 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC13 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC13 flag bit to 1. This will cause a conversion to start once priority is given to SOC13. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC13 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
12	SOC12	R-0/W1S	0h	SOC12 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC12 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC12 flag bit to 1. This will cause a conversion to start once priority is given to SOC12. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC12 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
11	SOC11	R-0/W1S	0h	SOC11 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC11 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC11 flag bit to 1. This will cause a conversion to start once priority is given to SOC11. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC11 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
10	SOC10	R-0/W1S	0h	SOC10 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC10 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC10 flag bit to 1. This will cause a conversion to start once priority is given to SOC10. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC10 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn

Table 24-95. ADCSOCFRC1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	SOC9	R-0/W1S	0h	SOC9 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC9 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC9 flag bit to 1. This will cause a conversion to start once priority is given to SOC9. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC9 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
8	SOC8	R-0/W1S	0h	SOC8 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC8 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC8 flag bit to 1. This will cause a conversion to start once priority is given to SOC8. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC8 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
7	SOC7	R-0/W1S	0h	SOC7 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC7 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC7 flag bit to 1. This will cause a conversion to start once priority is given to SOC7. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC7 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
6	SOC6	R-0/W1S	0h	SOC6 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC6 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC6 flag bit to 1. This will cause a conversion to start once priority is given to SOC6. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC6 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn

Table 24-95. ADCSOCFRC1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	SOC5	R-0/W1S	0h	SOC5 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC5 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC5 flag bit to 1. This will cause a conversion to start once priority is given to SOC5. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC5 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
4	SOC4	R-0/W1S	0h	SOC4 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC4 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC4 flag bit to 1. This will cause a conversion to start once priority is given to SOC4. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC4 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
3	SOC3	R-0/W1S	0h	SOC3 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC3 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC3 flag bit to 1. This will cause a conversion to start once priority is given to SOC3. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC3 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
2	SOC2	R-0/W1S	0h	SOC2 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC2 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC2 flag bit to 1. This will cause a conversion to start once priority is given to SOC2. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC2 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn

Table 24-95. ADCSOCFRC1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	SOC1	R-0/W1S	0h	SOC1 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC1 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC1 flag bit to 1. This will cause a conversion to start once priority is given to SOC1. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC1 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
0	SOC0	R-0/W1S	0h	SOC0 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC0 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC0 flag bit to 1. This will cause a conversion to start once priority is given to SOC0. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC0 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn

24.16.3.15 ADCSOCOVF1 Register (Offset = 2Ch) [Reset = 00000000h]

ADCSOCOVF1 is shown in Figure 24-118 and described in Table 24-96.

Return to the Summary Table.

ADC SOC Overflow 1 Register

Figure 24-118. ADCSOCOVF1 Register

31	30	29	28	27	26	25	24
SOC31	SOC30	SOC29	SOC28	SOC27	SOC26	SOC25	SOC24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
SOC23	SOC22	SOC21	SOC20	SOC19	SOC18	SOC17	SOC16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
SOC7	SOC6	SOC5	SOC4	SOC3	SOC2	SOC1	SOC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 24-96. ADCSOCOVF1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SOC31	R	0h	SOC31 Start of Conversion Overflow Flag. Indicates an SOC31 event was generated in hardware while an existing SOC31 event was already pending. 0 No SOC31 event overflow. 1 SOC31 event overflow. An overflow condition does not stop SOC31 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
30	SOC30	R	0h	SOC30 Start of Conversion Overflow Flag. Indicates an SOC30 event was generated in hardware while an existing SOC30 event was already pending. 0 No SOC30 event overflow. 1 SOC30 event overflow. An overflow condition does not stop SOC30 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
29	SOC29	R	0h	SOC29 Start of Conversion Overflow Flag. Indicates an SOC29 event was generated in hardware while an existing SOC29 event was already pending. 0 No SOC29 event overflow. 1 SOC29 event overflow. An overflow condition does not stop SOC29 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn

Table 24-96. ADCSOCOVF1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	SOC28	R	0h	SOC28 Start of Conversion Overflow Flag. Indicates an SOC28 event was generated in hardware while an existing SOC28 event was already pending. 0 No SOC28 event overflow. 1 SOC28 event overflow. An overflow condition does not stop SOC28 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
27	SOC27	R	0h	SOC27 Start of Conversion Overflow Flag. Indicates an SOC27 event was generated in hardware while an existing SOC27 event was already pending. 0 No SOC27 event overflow. 1 SOC27 event overflow. An overflow condition does not stop SOC27 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
26	SOC26	R	0h	SOC26 Start of Conversion Overflow Flag. Indicates an SOC26 event was generated in hardware while an existing SOC26 event was already pending. 0 No SOC26 event overflow. 1 SOC26 event overflow. An overflow condition does not stop SOC26 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
25	SOC25	R	0h	SOC25 Start of Conversion Overflow Flag. Indicates an SOC25 event was generated in hardware while an existing SOC25 event was already pending. 0 No SOC25 event overflow. 1 SOC25 event overflow. An overflow condition does not stop SOC25 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
24	SOC24	R	0h	SOC24 Start of Conversion Overflow Flag. Indicates an SOC24 event was generated in hardware while an existing SOC24 event was already pending. 0 No SOC24 event overflow. 1 SOC24 event overflow. An overflow condition does not stop SOC24 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
23	SOC23	R	0h	SOC23 Start of Conversion Overflow Flag. Indicates an SOC23 event was generated in hardware while an existing SOC23 event was already pending. 0 No SOC23 event overflow. 1 SOC23 event overflow. An overflow condition does not stop SOC23 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn

Table 24-96. ADCSOCOVF1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	SOC22	R	0h	SOC22 Start of Conversion Overflow Flag. Indicates an SOC22 event was generated in hardware while an existing SOC22 event was already pending. 0 No SOC22 event overflow. 1 SOC22 event overflow. An overflow condition does not stop SOC22 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
21	SOC21	R	0h	SOC21 Start of Conversion Overflow Flag. Indicates an SOC21 event was generated in hardware while an existing SOC21 event was already pending. 0 No SOC21 event overflow. 1 SOC21 event overflow. An overflow condition does not stop SOC21 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
20	SOC20	R	0h	SOC20 Start of Conversion Overflow Flag. Indicates an SOC20 event was generated in hardware while an existing SOC20 event was already pending. 0 No SOC20 event overflow. 1 SOC20 event overflow. An overflow condition does not stop SOC20 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
19	SOC19	R	0h	SOC19 Start of Conversion Overflow Flag. Indicates an SOC19 event was generated in hardware while an existing SOC19 event was already pending. 0 No SOC19 event overflow. 1 SOC19 event overflow. An overflow condition does not stop SOC19 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
18	SOC18	R	0h	SOC18 Start of Conversion Overflow Flag. Indicates an SOC18 event was generated in hardware while an existing SOC18 event was already pending. 0 No SOC18 event overflow. 1 SOC18 event overflow. An overflow condition does not stop SOC18 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
17	SOC17	R	0h	SOC17 Start of Conversion Overflow Flag. Indicates an SOC17 event was generated in hardware while an existing SOC17 event was already pending. 0 No SOC17 event overflow. 1 SOC17 event overflow. An overflow condition does not stop SOC17 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn

Table 24-96. ADCSOCOVF1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	SOC16	R	0h	SOC16 Start of Conversion Overflow Flag. Indicates an SOC16 event was generated in hardware while an existing SOC16 event was already pending. 0 No SOC16 event overflow. 1 SOC16 event overflow. An overflow condition does not stop SOC16 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
15	SOC15	R	0h	SOC15 Start of Conversion Overflow Flag. Indicates an SOC15 event was generated in hardware while an existing SOC15 event was already pending. 0 No SOC15 event overflow. 1 SOC15 event overflow. An overflow condition does not stop SOC15 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
14	SOC14	R	0h	SOC14 Start of Conversion Overflow Flag. Indicates an SOC14 event was generated in hardware while an existing SOC14 event was already pending. 0 No SOC14 event overflow. 1 SOC14 event overflow. An overflow condition does not stop SOC14 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
13	SOC13	R	0h	SOC13 Start of Conversion Overflow Flag. Indicates an SOC13 event was generated in hardware while an existing SOC13 event was already pending. 0 No SOC13 event overflow. 1 SOC13 event overflow. An overflow condition does not stop SOC13 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
12	SOC12	R	0h	SOC12 Start of Conversion Overflow Flag. Indicates an SOC12 event was generated in hardware while an existing SOC12 event was already pending. 0 No SOC12 event overflow. 1 SOC12 event overflow. An overflow condition does not stop SOC12 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
11	SOC11	R	0h	SOC11 Start of Conversion Overflow Flag. Indicates an SOC11 event was generated in hardware while an existing SOC11 event was already pending. 0 No SOC11 event overflow. 1 SOC11 event overflow. An overflow condition does not stop SOC11 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn

Table 24-96. ADCSOCOVF1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SOC10	R	0h	SOC10 Start of Conversion Overflow Flag. Indicates an SOC10 event was generated in hardware while an existing SOC10 event was already pending. 0 No SOC10 event overflow. 1 SOC10 event overflow. An overflow condition does not stop SOC10 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
9	SOC9	R	0h	SOC9 Start of Conversion Overflow Flag. Indicates an SOC9 event was generated in hardware while an existing SOC9 event was already pending. 0 No SOC9 event overflow. 1 SOC9 event overflow. An overflow condition does not stop SOC9 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
8	SOC8	R	0h	SOC8 Start of Conversion Overflow Flag. Indicates an SOC8 event was generated in hardware while an existing SOC8 event was already pending. 0 No SOC8 event overflow. 1 SOC8 event overflow. An overflow condition does not stop SOC8 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
7	SOC7	R	0h	SOC7 Start of Conversion Overflow Flag. Indicates an SOC7 event was generated in hardware while an existing SOC7 event was already pending. 0 No SOC7 event overflow. 1 SOC7 event overflow. An overflow condition does not stop SOC7 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
6	SOC6	R	0h	SOC6 Start of Conversion Overflow Flag. Indicates an SOC6 event was generated in hardware while an existing SOC6 event was already pending. 0 No SOC6 event overflow. 1 SOC6 event overflow. An overflow condition does not stop SOC6 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
5	SOC5	R	0h	SOC5 Start of Conversion Overflow Flag. Indicates an SOC5 event was generated in hardware while an existing SOC5 event was already pending. 0 No SOC5 event overflow. 1 SOC5 event overflow. An overflow condition does not stop SOC5 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn

Table 24-96. ADCSOCOVF1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	SOC4	R	0h	SOC4 Start of Conversion Overflow Flag. Indicates an SOC4 event was generated in hardware while an existing SOC4 event was already pending. 0 No SOC4 event overflow. 1 SOC4 event overflow. An overflow condition does not stop SOC4 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
3	SOC3	R	0h	SOC3 Start of Conversion Overflow Flag. Indicates an SOC3 event was generated in hardware while an existing SOC3 event was already pending. 0 No SOC3 event overflow. 1 SOC3 event overflow. An overflow condition does not stop SOC3 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
2	SOC2	R	0h	SOC2 Start of Conversion Overflow Flag. Indicates an SOC2 event was generated in hardware while an existing SOC2 event was already pending. 0 No SOC2 event overflow. 1 SOC2 event overflow. An overflow condition does not stop SOC2 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
1	SOC1	R	0h	SOC1 Start of Conversion Overflow Flag. Indicates an SOC1 event was generated in hardware while an existing SOC1 event was already pending. 0 No SOC1 event overflow. 1 SOC1 event overflow. An overflow condition does not stop SOC1 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
0	SOC0	R	0h	SOC0 Start of Conversion Overflow Flag. Indicates an SOC0 event was generated in hardware while an existing SOC0 event was already pending. 0 No SOC0 event overflow. 1 SOC0 event overflow. An overflow condition does not stop SOC0 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn

24.16.3.16 ADCSOCOVFCLR1 Register (Offset = 30h) [Reset = 00000000h]

ADCSOCOVFCLR1 is shown in Figure 24-119 and described in Table 24-97.

Return to the Summary Table.

ADC SOC Overflow Clear 1 Register

Figure 24-119. ADCSOCOVFCLR1 Register

31	30	29	28	27	26	25	24
SOC31	SOC30	SOC29	SOC28	SOC27	SOC26	SOC25	SOC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
SOC23	SOC22	SOC21	SOC20	SOC19	SOC18	SOC17	SOC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
SOC7	SOC6	SOC5	SOC4	SOC3	SOC2	SOC1	SOC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 24-97. ADCSOCOVFCLR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SOC31	R-0/W1S	0h	SOC31 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC31 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC31 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
30	SOC30	R-0/W1S	0h	SOC30 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC30 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC30 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
29	SOC29	R-0/W1S	0h	SOC29 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC29 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC29 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

Table 24-97. ADCSOCOVFCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	SOC28	R-0/W1S	0h	SOC28 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC28 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC28 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
27	SOC27	R-0/W1S	0h	SOC27 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC27 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC27 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
26	SOC26	R-0/W1S	0h	SOC26 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC26 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC26 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
25	SOC25	R-0/W1S	0h	SOC25 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC25 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC25 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
24	SOC24	R-0/W1S	0h	SOC24 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC24 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC24 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
23	SOC23	R-0/W1S	0h	SOC23 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC23 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC23 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

Table 24-97. ADCSOCOVFCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	SOC22	R-0/W1S	0h	SOC22 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC22 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC22 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
21	SOC21	R-0/W1S	0h	SOC21 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC21 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC21 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
20	SOC20	R-0/W1S	0h	SOC20 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC20 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC20 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
19	SOC19	R-0/W1S	0h	SOC19 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC19 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC19 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
18	SOC18	R-0/W1S	0h	SOC18 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC18 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC18 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
17	SOC17	R-0/W1S	0h	SOC17 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC17 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC17 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

Table 24-97. ADCSOCOVFCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	SOC16	R-0/W1S	0h	SOC16 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC16 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC16 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
15	SOC15	R-0/W1S	0h	SOC15 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC15 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC15 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
14	SOC14	R-0/W1S	0h	SOC14 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC14 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC14 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
13	SOC13	R-0/W1S	0h	SOC13 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC13 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC13 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
12	SOC12	R-0/W1S	0h	SOC12 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC12 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC12 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
11	SOC11	R-0/W1S	0h	SOC11 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC11 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC11 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

Table 24-97. ADCSOCOVFCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SOC10	R-0/W1S	0h	SOC10 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC10 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC10 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
9	SOC9	R-0/W1S	0h	SOC9 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC9 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC9 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
8	SOC8	R-0/W1S	0h	SOC8 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC8 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC8 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
7	SOC7	R-0/W1S	0h	SOC7 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC7 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC7 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
6	SOC6	R-0/W1S	0h	SOC6 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC6 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC6 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
5	SOC5	R-0/W1S	0h	SOC5 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC5 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC5 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

Table 24-97. ADCSOCOVFCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	SOC4	R-0/W1S	0h	SOC4 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC4 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC4 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
3	SOC3	R-0/W1S	0h	SOC3 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC3 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC3 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
2	SOC2	R-0/W1S	0h	SOC2 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC2 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC2 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
1	SOC1	R-0/W1S	0h	SOC1 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC1 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC1 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
0	SOC0	R-0/W1S	0h	SOC0 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC0 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC0 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

24.16.3.17 ADCSOC0CTL Register (Offset = 34h) [Reset = 00000000h]

ADCSOC0CTL is shown in Figure 24-120 and described in Table 24-98.

Return to the Summary Table.

ADC SOC0 Control Register

Figure 24-120. ADCSOC0CTL Register

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

Table 24-98. ADCSOC0CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC0 External Channel Mux Select. Selects the external mux combination to output when SOC0 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC0 Trigger Source Select. Along with the SOC0 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC0 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-98. ADCSOC0CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC0 Channel Select. Selects the channel to be converted when SOC0 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC0 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.18 ADCSOC1CTL Register (Offset = 38h) [Reset = 00000000h]

ADCSOC1CTL is shown in Figure 24-121 and described in Table 24-99.

Return to the Summary Table.

ADC SOC1 Control Register

Figure 24-121. ADCSOC1CTL Register

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

Table 24-99. ADCSOC1CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC1 External Channel Mux Select. Selects the external mux combination to output when SOC1 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC1 Trigger Source Select. Along with the SOC1 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC1 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-99. ADCSOC1CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC1 Channel Select. Selects the channel to be converted when SOC1 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC1 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.19 ADCSOC2CTL Register (Offset = 3Ch) [Reset = 00000000h]

ADCSOC2CTL is shown in Figure 24-122 and described in Table 24-100.

Return to the Summary Table.

ADC SOC2 Control Register

Figure 24-122. ADCSOC2CTL Register

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

Table 24-100. ADCSOC2CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC2 External Channel Mux Select. Selects the external mux combination to output when SOC2 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC2 Trigger Source Select. Along with the SOC2 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC2 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-100. ADCSOC2CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC2 Channel Select. Selects the channel to be converted when SOC2 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC2 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.20 ADCSOC3CTL Register (Offset = 40h) [Reset = 00000000h]

ADCSOC3CTL is shown in Figure 24-123 and described in Table 24-101.

Return to the Summary Table.

ADC SOC3 Control Register

Figure 24-123. ADCSOC3CTL Register

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

Table 24-101. ADCSOC3CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC3 External Channel Mux Select. Selects the external mux combination to output when SOC3 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC3 Trigger Source Select. Along with the SOC3 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC3 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-101. ADCSOC3CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC3 Channel Select. Selects the channel to be converted when SOC3 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC3 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.21 ADCSOC4CTL Register (Offset = 44h) [Reset = 00000000h]

ADCSOC4CTL is shown in Figure 24-124 and described in Table 24-102.

Return to the Summary Table.

ADC SOC4 Control Register

Figure 24-124. ADCSOC4CTL Register

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

Table 24-102. ADCSOC4CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC4 External Channel Mux Select. Selects the external mux combination to output when SOC4 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC4 Trigger Source Select. Along with the SOC4 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC4 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-102. ADCSOC4CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC4 Channel Select. Selects the channel to be converted when SOC4 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC4 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.22 ADCSOC5CTL Register (Offset = 48h) [Reset = 00000000h]

ADCSOC5CTL is shown in Figure 24-125 and described in Table 24-103.

Return to the Summary Table.

ADC SOC5 Control Register

Figure 24-125. ADCSOC5CTL Register

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

Table 24-103. ADCSOC5CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC5 External Channel Mux Select. Selects the external mux combination to output when SOC5 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC5 Trigger Source Select. Along with the SOC5 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC5 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-103. ADCSOC5CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC5 Channel Select. Selects the channel to be converted when SOC5 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC5 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.23 ADCSOC6CTL Register (Offset = 4Ch) [Reset = 00000000h]

ADCSOC6CTL is shown in Figure 24-126 and described in Table 24-104.

Return to the Summary Table.

ADC SOC6 Control Register

Figure 24-126. ADCSOC6CTL Register

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

Table 24-104. ADCSOC6CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC6 External Channel Mux Select. Selects the external mux combination to output when SOC6 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC6 Trigger Source Select. Along with the SOC6 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC6 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-104. ADCSOC6CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC6 Channel Select. Selects the channel to be converted when SOC6 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC6 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.24 ADCSOC7CTL Register (Offset = 50h) [Reset = 00000000h]

ADCSOC7CTL is shown in Figure 24-127 and described in Table 24-105.

Return to the Summary Table.

ADC SOC7 Control Register

Figure 24-127. ADCSOC7CTL Register

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

Table 24-105. ADCSOC7CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC7 External Channel Mux Select. Selects the external mux combination to output when SOC7 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC7 Trigger Source Select. Along with the SOC7 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC7 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-105. ADCSOC7CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC7 Channel Select. Selects the channel to be converted when SOC7 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC7 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.25 ADCSOC8CTL Register (Offset = 54h) [Reset = 00000000h]

ADCSOC8CTL is shown in Figure 24-128 and described in Table 24-106.

Return to the Summary Table.

ADC SOC8 Control Register

Figure 24-128. ADCSOC8CTL Register

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

Table 24-106. ADCSOC8CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC8 External Channel Mux Select. Selects the external mux combination to output when SOC8 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC8 Trigger Source Select. Along with the SOC8 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC8 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-106. ADCSOC8CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC8 Channel Select. Selects the channel to be converted when SOC8 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC8 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.26 ADCSOC9CTL Register (Offset = 58h) [Reset = 00000000h]

ADCSOC9CTL is shown in Figure 24-129 and described in Table 24-107.

Return to the Summary Table.

ADC SOC9 Control Register

Figure 24-129. ADCSOC9CTL Register

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

Table 24-107. ADCSOC9CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC9 External Channel Mux Select. Selects the external mux combination to output when SOC9 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC9 Trigger Source Select. Along with the SOC9 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC9 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-107. ADCSOC9CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC9 Channel Select. Selects the channel to be converted when SOC9 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC9 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.27 ADCSOC10CTL Register (Offset = 5Ch) [Reset = 00000000h]

ADCSOC10CTL is shown in Figure 24-130 and described in Table 24-108.

Return to the Summary Table.

ADC SOC10 Control Register

Figure 24-130. ADCSOC10CTL Register

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

Table 24-108. ADCSOC10CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC10 External Channel Mux Select. Selects the external mux combination to output when SOC10 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC10 Trigger Source Select. Along with the SOC10 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC10 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-108. ADCSOC10CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC10 Channel Select. Selects the channel to be converted when SOC10 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC10 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.28 ADCSOC11CTL Register (Offset = 60h) [Reset = 00000000h]

ADCSOC11CTL is shown in Figure 24-131 and described in Table 24-109.

Return to the Summary Table.

ADC SOC11 Control Register

Figure 24-131. ADCSOC11CTL Register

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

Table 24-109. ADCSOC11CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC11 External Channel Mux Select. Selects the external mux combination to output when SOC11 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC11 Trigger Source Select. Along with the SOC11 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC11 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-109. ADCSOC11CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC11 Channel Select. Selects the channel to be converted when SOC11 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC11 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.29 ADCSOC12CTL Register (Offset = 64h) [Reset = 00000000h]

ADCSOC12CTL is shown in Figure 24-132 and described in Table 24-110.

Return to the Summary Table.

ADC SOC12 Control Register

Figure 24-132. ADCSOC12CTL Register

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

Table 24-110. ADCSOC12CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC12 External Channel Mux Select. Selects the external mux combination to output when SOC12 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC12 Trigger Source Select. Along with the SOC12 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC12 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-110. ADCSOC12CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC12 Channel Select. Selects the channel to be converted when SOC12 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC12 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.30 ADCSOC13CTL Register (Offset = 68h) [Reset = 00000000h]

ADCSOC13CTL is shown in Figure 24-133 and described in Table 24-111.

Return to the Summary Table.

ADC SOC13 Control Register

Figure 24-133. ADCSOC13CTL Register

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

Table 24-111. ADCSOC13CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC13 External Channel Mux Select. Selects the external mux combination to output when SOC13 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC13 Trigger Source Select. Along with the SOC13 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC13 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-111. ADCSOC13CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC13 Channel Select. Selects the channel to be converted when SOC13 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC13 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.31 ADCSOC14CTL Register (Offset = 6Ch) [Reset = 00000000h]

ADCSOC14CTL is shown in Figure 24-134 and described in Table 24-112.

Return to the Summary Table.

ADC SOC14 Control Register

Figure 24-134. ADCSOC14CTL Register

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

Table 24-112. ADCSOC14CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC14 External Channel Mux Select. Selects the external mux combination to output when SOC14 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC14 Trigger Source Select. Along with the SOC14 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC14 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-112. ADCSOC14CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC14 Channel Select. Selects the channel to be converted when SOC14 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC14 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.32 ADCSOC15CTL Register (Offset = 70h) [Reset = 00000000h]

ADCSOC15CTL is shown in Figure 24-135 and described in Table 24-113.

Return to the Summary Table.

ADC SOC15 Control Register

Figure 24-135. ADCSOC15CTL Register

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

Table 24-113. ADCSOC15CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC15 External Channel Mux Select. Selects the external mux combination to output when SOC15 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC15 Trigger Source Select. Along with the SOC15 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC15 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-113. ADCSOC15CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC15 Channel Select. Selects the channel to be converted when SOC15 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC15 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.33 ADCSOC16CTL Register (Offset = 74h) [Reset = 00000000h]

ADCSOC16CTL is shown in Figure 24-136 and described in Table 24-114.

Return to the Summary Table.

ADC SOC16 Control Register

Figure 24-136. ADCSOC16CTL Register

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

Table 24-114. ADCSOC16CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC16 External Channel Mux Select. Selects the external mux combination to output when SOC16 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC16 Trigger Source Select. Along with the SOC16 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC16 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-114. ADCSOC16CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC16 Channel Select. Selects the channel to be converted when SOC16 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC16 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.34 ADCSOC17CTL Register (Offset = 78h) [Reset = 00000000h]

ADCSOC17CTL is shown in Figure 24-137 and described in Table 24-115.

Return to the Summary Table.

ADC SOC17 Control Register

Figure 24-137. ADCSOC17CTL Register

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

Table 24-115. ADCSOC17CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC17 External Channel Mux Select. Selects the external mux combination to output when SOC17 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC17 Trigger Source Select. Along with the SOC17 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC17 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-115. ADCSOC17CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC17 Channel Select. Selects the channel to be converted when SOC17 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC17 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.35 ADCSOC18CTL Register (Offset = 7Ch) [Reset = 00000000h]

ADCSOC18CTL is shown in Figure 24-138 and described in Table 24-116.

Return to the Summary Table.

ADC SOC18 Control Register

Figure 24-138. ADCSOC18CTL Register

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

Table 24-116. ADCSOC18CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC18 External Channel Mux Select. Selects the external mux combination to output when SOC18 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC18 Trigger Source Select. Along with the SOC18 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC18 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-116. ADCSOC18CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC18 Channel Select. Selects the channel to be converted when SOC18 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC18 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.36 ADCSOC19CTL Register (Offset = 80h) [Reset = 00000000h]

ADCSOC19CTL is shown in Figure 24-139 and described in Table 24-117.

Return to the Summary Table.

ADC SOC19 Control Register

Figure 24-139. ADCSOC19CTL Register

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

Table 24-117. ADCSOC19CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC19 External Channel Mux Select. Selects the external mux combination to output when SOC19 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC19 Trigger Source Select. Along with the SOC19 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC19 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-117. ADCSOC19CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC19 Channel Select. Selects the channel to be converted when SOC19 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC19 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.37 ADCSOC20CTL Register (Offset = 84h) [Reset = 00000000h]

ADCSOC20CTL is shown in Figure 24-140 and described in Table 24-118.

Return to the Summary Table.

ADC SOC20 Control Register

Figure 24-140. ADCSOC20CTL Register

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

Table 24-118. ADCSOC20CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC20 External Channel Mux Select. Selects the external mux combination to output when SOC20 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC20 Trigger Source Select. Along with the SOC20 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC20 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-118. ADCSOC20CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC20 Channel Select. Selects the channel to be converted when SOC20 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC20 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.38 ADCSOC21CTL Register (Offset = 88h) [Reset = 00000000h]

ADCSOC21CTL is shown in Figure 24-141 and described in Table 24-119.

Return to the Summary Table.

ADC SOC21 Control Register

Figure 24-141. ADCSOC21CTL Register

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

Table 24-119. ADCSOC21CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC21 External Channel Mux Select. Selects the external mux combination to output when SOC21 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC21 Trigger Source Select. Along with the SOC21 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC21 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-119. ADCSOC21CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC21 Channel Select. Selects the channel to be converted when SOC21 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC21 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.39 ADCSOC22CTL Register (Offset = 8Ch) [Reset = 00000000h]

ADCSOC22CTL is shown in Figure 24-142 and described in Table 24-120.

Return to the Summary Table.

ADC SOC22 Control Register

Figure 24-142. ADCSOC22CTL Register

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

Table 24-120. ADCSOC22CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC22 External Channel Mux Select. Selects the external mux combination to output when SOC22 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC22 Trigger Source Select. Along with the SOC22 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC22 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-120. ADCSOC22CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC22 Channel Select. Selects the channel to be converted when SOC22 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC22 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.40 ADCSOC23CTL Register (Offset = 90h) [Reset = 00000000h]

ADCSOC23CTL is shown in Figure 24-143 and described in Table 24-121.

Return to the Summary Table.

ADC SOC23 Control Register

Figure 24-143. ADCSOC23CTL Register

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

Table 24-121. ADCSOC23CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC23 External Channel Mux Select. Selects the external mux combination to output when SOC23 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC23 Trigger Source Select. Along with the SOC23 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC23 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-121. ADCSOC23CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC23 Channel Select. Selects the channel to be converted when SOC23 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC23 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.41 ADCSOC24CTL Register (Offset = 94h) [Reset = 00000000h]

ADCSOC24CTL is shown in Figure 24-144 and described in Table 24-122.

Return to the Summary Table.

ADC SOC24 Control Register

Figure 24-144. ADCSOC24CTL Register

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

Table 24-122. ADCSOC24CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC24 External Channel Mux Select. Selects the external mux combination to output when SOC24 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC24 Trigger Source Select. Along with the SOC24 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC24 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-122. ADCSOC24CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC24 Channel Select. Selects the channel to be converted when SOC24 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC24 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.42 ADCSOC25CTL Register (Offset = 98h) [Reset = 00000000h]

ADCSOC25CTL is shown in Figure 24-145 and described in Table 24-123.

Return to the Summary Table.

ADC SOC25 Control Register

Figure 24-145. ADCSOC25CTL Register

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

Table 24-123. ADCSOC25CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC25 External Channel Mux Select. Selects the external mux combination to output when SOC25 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC25 Trigger Source Select. Along with the SOC25 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC25 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-123. ADCSOC25CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC25 Channel Select. Selects the channel to be converted when SOC25 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC25 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.43 ADCSOC26CTL Register (Offset = 9Ch) [Reset = 00000000h]

ADCSOC26CTL is shown in Figure 24-146 and described in Table 24-124.

Return to the Summary Table.

ADC SOC26 Control Register

Figure 24-146. ADCSOC26CTL Register

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

Table 24-124. ADCSOC26CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC26 External Channel Mux Select. Selects the external mux combination to output when SOC26 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC26 Trigger Source Select. Along with the SOC26 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC26 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-124. ADCSOC26CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC26 Channel Select. Selects the channel to be converted when SOC26 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC26 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.44 ADCSOC27CTL Register (Offset = A0h) [Reset = 00000000h]

ADCSOC27CTL is shown in Figure 24-147 and described in Table 24-125.

Return to the Summary Table.

ADC SOC27 Control Register

Figure 24-147. ADCSOC27CTL Register

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

Table 24-125. ADCSOC27CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC27 External Channel Mux Select. Selects the external mux combination to output when SOC27 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC27 Trigger Source Select. Along with the SOC27 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC27 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-125. ADCSOC27CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC27 Channel Select. Selects the channel to be converted when SOC27 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC27 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.45 ADCSOC28CTL Register (Offset = A4h) [Reset = 00000000h]

ADCSOC28CTL is shown in Figure 24-148 and described in Table 24-126.

Return to the Summary Table.

ADC SOC28 Control Register

Figure 24-148. ADCSOC28CTL Register

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

Table 24-126. ADCSOC28CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC28 External Channel Mux Select. Selects the external mux combination to output when SOC28 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC28 Trigger Source Select. Along with the SOC28 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC28 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-126. ADCSOC28CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC28 Channel Select. Selects the channel to be converted when SOC28 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC28 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.46 ADCSOC29CTL Register (Offset = A8h) [Reset = 00000000h]

ADCSOC29CTL is shown in Figure 24-149 and described in Table 24-127.

Return to the Summary Table.

ADC SOC29 Control Register

Figure 24-149. ADCSOC29CTL Register

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

Table 24-127. ADCSOC29CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC29 External Channel Mux Select. Selects the external mux combination to output when SOC29 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC29 Trigger Source Select. Along with the SOC29 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC29 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-127. ADCSOC29CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC29 Channel Select. Selects the channel to be converted when SOC29 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC29 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.47 ADCSOC30CTL Register (Offset = ACh) [Reset = 00000000h]

ADCSOC30CTL is shown in Figure 24-150 and described in Table 24-128.

Return to the Summary Table.

ADC SOC30 Control Register

Figure 24-150. ADCSOC30CTL Register

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

Table 24-128. ADCSOC30CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC30 External Channel Mux Select. Selects the external mux combination to output when SOC30 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC30 Trigger Source Select. Along with the SOC30 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC30 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-128. ADCSOC30CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC30 Channel Select. Selects the channel to be converted when SOC30 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC30 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.48 ADCSOC31CTL Register (Offset = B0h) [Reset = 00000000h]

ADCSOC31CTL is shown in Figure 24-151 and described in Table 24-129.

Return to the Summary Table.

ADC SOC31 Control Register

Figure 24-151. ADCSOC31CTL Register

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

Table 24-129. ADCSOC31CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	EXTCHSEL	R/W	0h	SOC31 External Channel Mux Select. Selects the external mux combination to output when SOC31 is received by the ADC. Some or all of the ADCEXTMUX lines can be enabled via the device GPIO mux to control an external analog mux. 0h ADCEXTMUX[3:0] = 0000 1h ADCEXTMUX[3:0] = 0001 2h ADCEXTMUX[3:0] = 0010 3h ADCEXTMUX[3:0] = 0011 4h ADCEXTMUX[3:0] = 0100 5h ADCEXTMUX[3:0] = 0101 6h ADCEXTMUX[3:0] = 0110 7h ADCEXTMUX[3:0] = 0111 8h ADCEXTMUX[3:0] = 1000 9h ADCEXTMUX[3:0] = 1001 Ah ADCEXTMUX[3:0] = 1010 Bh ADCEXTMUX[3:0] = 1011 Ch ADCEXTMUX[3:0] = 1100 Dh ADCEXTMUX[3:0] = 1101 Eh ADCEXTMUX[3:0] = 1110 Fh ADCEXTMUX[3:0] = 1111 Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26-20	TRIGSEL	R/W	0h	SOC31 Trigger Source Select. Along with the SOC31 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC31 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h-7fh ADCTRIG1-ADCTRG127 - Hardware trigger sources Reset type: SYSRSn

Table 24-129. ADCSOC31CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-15	CHSEL	R/W	0h	SOC31 Channel Select. Selects the channel to be converted when SOC31 is received by the ADC. Single-ended Signaling Mode (SIGNALMODE = 0): 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Differential Signaling Mode (SIGNALMODE = 1): 00h ADCIN0 (non-inverting) and ADCIN1 (inverting) 01h ADCIN0 (non-inverting) and ADCIN1 (inverting) 02h ADCIN2 (non-inverting) and ADCIN3 (inverting) 03h ADCIN2 (non-inverting) and ADCIN3 (inverting) 04h ADCIN4 (non-inverting) and ADCIN5 (inverting) 05h ADCIN4 (non-inverting) and ADCIN5 (inverting) ... 0Eh ADCIN26 (non-inverting) and ADCIN27 (inverting) 0Fh ADCIN26 (non-inverting) and ADCIN27 (inverting) 10h ADCIN28 (non-inverting) and ADCIN29 (inverting) 11h ADCIN28 (non-inverting) and ADCIN29 (inverting) 1Eh ADCIN30 (non-inverting) and ADCIN31 (inverting) 1Fh ADCIN30 (non-inverting) and ADCIN31 (inverting) Reset type: SYSRSn
14-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	0h	Reserved
8-0	ACQPS	R/W	0h	SOC31 Acquisition Prescale. Controls the sample and hold window for this SOC. 000h Reserved 001h Reserved 002h Sample window is 3 system clock cycles wide 003h Sample window is 4 system clock cycles wide ... 1FFh Sample window is 512 system clock cycles wide The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The configured acquisition time must also be at least 3 SYSCLK cycles long. The device datasheet will also specify a minimum sample and hold window duration. Reset type: SYSRSn

24.16.3.49 ADCEVTSTAT Register (Offset = B4h) [Reset = 0000h]

ADCEVTSTAT is shown in Figure 24-152 and described in Table 24-130.

Return to the Summary Table.

ADC Event Status Register

Figure 24-152. ADCEVTSTAT Register

15	14	13	12	11	10	9	8
RESERVED	PPB4ZERO	PPB4TRIPLO	PPB4TRIPHI	RESERVED	PPB3ZERO	PPB3TRIPLO	PPB3TRIPHI
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	PPB2ZERO	PPB2TRIPLO	PPB2TRIPHI	RESERVED	PPB1ZERO	PPB1TRIPLO	PPB1TRIPHI
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 24-130. ADCEVTSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	PPB4ZERO	R	0h	Post Processing Block 4 Zero Crossing Flag. When set indicates the ADCPPB4RESULT register has changed sign. This bit is gated by EOC signal. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
13	PPB4TRIPLO	R	0h	Post Processing Block 4 Trip Low Flag. When set indicates a digital compare trip low event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
12	PPB4TRIPHI	R	0h	Post Processing Block 4 Trip High Flag. When set indicates a digital compare trip high event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
11	RESERVED	R	0h	Reserved
10	PPB3ZERO	R	0h	Post Processing Block 3 Zero Crossing Flag. When set indicates the ADCPPB3RESULT register has changed sign. This bit is gated by EOC signal. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

Table 24-130. ADCEVTSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	PPB3TRIPLO	R	0h	Post Processing Block 3 Trip Low Flag. When set indicates a digital compare trip low event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
8	PPB3TRIPHI	R	0h	Post Processing Block 3 Trip High Flag. When set indicates a digital compare trip high event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6	PPB2ZERO	R	0h	Post Processing Block 2 Zero Crossing Flag. When set indicates the ADCPPB2RESULT register has changed sign. This bit is gated by EOC signal. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
5	PPB2TRIPLO	R	0h	Post Processing Block 2 Trip Low Flag. When set indicates a digital compare trip low event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
4	PPB2TRIPHI	R	0h	Post Processing Block 2 Trip High Flag. When set indicates a digital compare trip high event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
3	RESERVED	R	0h	Reserved
2	PPB1ZERO	R	0h	Post Processing Block 1 Zero Crossing Flag. When set indicates the ADCPPB1RESULT register has changed sign. This bit is gated by EOC signal. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

Table 24-130. ADCEVTSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	PPB1TRIPLO	R	0h	Post Processing Block 1 Trip Low Flag. When set indicates a digital compare trip low event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
0	PPB1TRIPHI	R	0h	Post Processing Block 1 Trip High Flag. When set indicates a digital compare trip high event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

24.16.3.50 ADCEVTCLR Register (Offset = B8h) [Reset = 0000h]

ADCEVTCLR is shown in Figure 24-153 and described in Table 24-131.

Return to the Summary Table.

ADC Event Clear Register

Figure 24-153. ADCEVTCLR Register

15	14	13	12	11	10	9	8
RESERVED	PPB4ZERO	PPB4TRIPLO	PPB4TRIPHI	RESERVED	PPB3ZERO	PPB3TRIPLO	PPB3TRIPHI
R-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	PPB2ZERO	PPB2TRIPLO	PPB2TRIPHI	RESERVED	PPB1ZERO	PPB1TRIPLO	PPB1TRIPHI
R-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 24-131. ADCEVTCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	PPB4ZERO	R-0/W1S	0h	Post Processing Block 4 Zero Crossing Clear. Clears the corresponding zero crossing flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
13	PPB4TRIPLO	R-0/W1S	0h	Post Processing Block 4 Trip Low Clear. Clears the corresponding trip low flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
12	PPB4TRIPHI	R-0/W1S	0h	Post Processing Block 4 Trip High Clear. Clears the corresponding trip high flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
11	RESERVED	R	0h	Reserved
10	PPB3ZERO	R-0/W1S	0h	Post Processing Block 3 Zero Crossing Clear. Clears the corresponding zero crossing flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
9	PPB3TRIPLO	R-0/W1S	0h	Post Processing Block 3 Trip Low Clear. Clears the corresponding trip low flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
8	PPB3TRIPHI	R-0/W1S	0h	Post Processing Block 3 Trip High Clear. Clears the corresponding trip high flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6	PPB2ZERO	R-0/W1S	0h	Post Processing Block 2 Zero Crossing Clear. Clears the corresponding zero crossing flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

Table 24-131. ADCEVTCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	PPB2TRIPLO	R-0/W1S	0h	Post Processing Block 2 Trip Low Clear. Clears the corresponding trip low flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
4	PPB2TRIPHI	R-0/W1S	0h	Post Processing Block 2 Trip High Clear. Clears the corresponding trip high flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
3	RESERVED	R	0h	Reserved
2	PPB1ZERO	R-0/W1S	0h	Post Processing Block 1 Zero Crossing Clear. Clears the corresponding zero crossing flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
1	PPB1TRIPLO	R-0/W1S	0h	Post Processing Block 1 Trip Low Clear. Clears the corresponding trip low flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
0	PPB1TRIPHI	R-0/W1S	0h	Post Processing Block 1 Trip High Clear. Clears the corresponding trip high flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

24.16.3.51 ADCEVTSEL Register (Offset = BCh) [Reset = 0000h]

ADCEVTSEL is shown in Figure 24-154 and described in Table 24-132.

Return to the Summary Table.

ADC Event Selection Register

Figure 24-154. ADCEVTSEL Register

15	14	13	12	11	10	9	8
RESERVED	PPB4ZERO	PPB4TRIPLO	PPB4TRIPHI	RESERVED	PPB3ZERO	PPB3TRIPLO	PPB3TRIPHI
R-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	PPB2ZERO	PPB2TRIPLO	PPB2TRIPHI	RESERVED	PPB1ZERO	PPB1TRIPLO	PPB1TRIPHI
R-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-132. ADCEVTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	PPB4ZERO	R/W	0h	Post Processing Block 4 Zero Crossing Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
13	PPB4TRIPLO	R/W	0h	Post Processing Block 4 Trip Low Event Enable. Setting this bit allows the corresponding rising trip low flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
12	PPB4TRIPHI	R/W	0h	Post Processing Block 4 Trip High Event Enable. Setting this bit allows the corresponding rising trip high flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
11	RESERVED	R	0h	Reserved
10	PPB3ZERO	R/W	0h	Post Processing Block 3 Zero Crossing Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
9	PPB3TRIPLO	R/W	0h	Post Processing Block 3 Trip Low Event Enable. Setting this bit allows the corresponding rising trip low flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
8	PPB3TRIPHI	R/W	0h	Post Processing Block 3 Trip High Event Enable. Setting this bit allows the corresponding rising trip high flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6	PPB2ZERO	R/W	0h	Post Processing Block 2 Zero Crossing Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn

Table 24-132. ADCEVTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	PPB2TRIPLO	R/W	0h	Post Processing Block 2 Trip Low Event Enable. Setting this bit allows the corresponding rising trip low flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
4	PPB2TRIPHI	R/W	0h	Post Processing Block 2 Trip High Event Enable. Setting this bit allows the corresponding rising trip high flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
3	RESERVED	R	0h	Reserved
2	PPB1ZERO	R/W	0h	Post Processing Block 1 Zero Crossing Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
1	PPB1TRIPLO	R/W	0h	Post Processing Block 1 Trip Low Event Enable. Setting this bit allows the corresponding rising trip low flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
0	PPB1TRIPHI	R/W	0h	Post Processing Block 1 Trip High Event Enable. Setting this bit allows the corresponding rising trip high flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn

24.16.3.52 ADCEVTINTSEL Register (Offset = C0h) [Reset = 0000h]

ADCEVTINTSEL is shown in Figure 24-155 and described in Table 24-133.

Return to the Summary Table.

ADC Event Interrupt Selection Register

Figure 24-155. ADCEVTINTSEL Register

15	14	13	12	11	10	9	8
RESERVED	PPB4ZERO	PPB4TRIPLO	PPB4TRIPHI	RESERVED	PPB3ZERO	PPB3TRIPLO	PPB3TRIPHI
R-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	PPB2ZERO	PPB2TRIPLO	PPB2TRIPHI	RESERVED	PPB1ZERO	PPB1TRIPLO	PPB1TRIPHI
R-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-133. ADCEVTINTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	PPB4ZERO	R/W	0h	Post Processing Block 4 Zero Crossing Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
13	PPB4TRIPLO	R/W	0h	Post Processing Block 4 Trip Low Interrupt Enable. Setting this bit allows the corresponding rising trip low flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
12	PPB4TRIPHI	R/W	0h	Post Processing Block 4 Trip High Interrupt Enable. Setting this bit allows the corresponding rising trip high flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
11	RESERVED	R	0h	Reserved
10	PPB3ZERO	R/W	0h	Post Processing Block 3 Zero Crossing Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
9	PPB3TRIPLO	R/W	0h	Post Processing Block 3 Trip Low Interrupt Enable. Setting this bit allows the corresponding rising trip low flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
8	PPB3TRIPHI	R/W	0h	Post Processing Block 3 Trip High Interrupt Enable. Setting this bit allows the corresponding rising trip high flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6	PPB2ZERO	R/W	0h	Post Processing Block 2 Zero Crossing Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn

Table 24-133. ADCEVTINTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	PPB2TRIPLO	R/W	0h	Post Processing Block 2 Trip Low Interrupt Enable. Setting this bit allows the corresponding rising trip low flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
4	PPB2TRIPHI	R/W	0h	Post Processing Block 2 Trip High Interrupt Enable. Setting this bit allows the corresponding rising trip high flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
3	RESERVED	R	0h	Reserved
2	PPB1ZERO	R/W	0h	Post Processing Block 1 Zero Crossing Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
1	PPB1TRIPLO	R/W	0h	Post Processing Block 1 Trip Low Interrupt Enable. Setting this bit allows the corresponding rising trip low flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
0	PPB1TRIPHI	R/W	0h	Post Processing Block 1 Trip High Interrupt Enable. Setting this bit allows the corresponding rising trip high flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn

24.16.3.53 ADCOSDETECT Register (Offset = C4h) [Reset = 0000h]

ADCOSDETECT is shown in Figure 24-156 and described in Table 24-134.

Return to the Summary Table.

ADC Open and Shorts Detect Register

Figure 24-156. ADCOSDETECT Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					DETECTCFG		
R-0h					R/W-0h		

Table 24-134. ADCOSDETECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-3	RESERVED	R	0h	Reserved
2-0	DETECTCFG	R/W	0h	ADC Opens and Shorts Detect Configuration. This bit field defines the open/shorts detection circuit state. 0h Open/Shorts detection circuit is disabled. 1h Open/Shorts detection circuit is enabled at zero scale. 2h Open/Shorts detection circuit is enabled at full scale. 3h Open/Shorts detection circuit is enabled at (nominal) 5/12 scale. 4h Open/Shorts detection circuit is enabled at (nominal) 7/12 scale. 5h Open/Shorts detection circuit is enabled with a (nominal) 5K pulldown to VSSA. 6h Open/Shorts detection circuit is enabled with a (nominal) 5K pullup to VDDA. 7h Open/Shorts detection circuit is enabled with a (nominal) 7K pulldown to VSSA. Reset type: SYSRSn

24.16.3.54 ADCCOUNTER Register (Offset = C6h) [Reset = 0000h]

ADCCOUNTER is shown in Figure 24-157 and described in Table 24-135.

Return to the Summary Table.

ADC Counter Register

Figure 24-157. ADCCOUNTER Register

15	14	13	12	11	10	9	8
RESERVED				FREECOUNT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
FREECOUNT							
R-0h							

Table 24-135. ADCCOUNTER Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	FREECOUNT	R	0h	ADC Free Running Counter Value. This bit field reflects the status of the free running ADC counter. Reset type: SYSRSn

24.16.3.55 ADCREV Register (Offset = C8h) [Reset = 0105h]

ADCREV is shown in Figure 24-158 and described in Table 24-136.

Return to the Summary Table.

ADC Revision Register

Figure 24-158. ADCREV Register

15	14	13	12	11	10	9	8
REV							
R-1h							
7	6	5	4	3	2	1	0
TYPE							
R-5h							

Table 24-136. ADCREV Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	REV	R	1h	ADC Revision. To allow documentation of differences between revisions. First version is labeled as 00h. Reset type: SYSRSn
7-0	TYPE	R	5h	ADC Type. Always set to 5 for this ADC. Reset type: SYSRSn

24.16.3.56 ADCOFFTRIM Register (Offset = CAh) [Reset = 0000h]

ADCOFFTRIM is shown in Figure 24-159 and described in Table 24-137.

Return to the Summary Table.

ADC Offset Trim Register 1

Figure 24-159. ADCOFFTRIM Register

15	14	13	12	11	10	9	8
OFFTRIM12BSEODD							
R/W-0h							
7	6	5	4	3	2	1	0
OFFTRIM							
R/W-0h							

Table 24-137. ADCOFFTRIM Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	OFFTRIM12BSEODD	R/W	0h	If ADCCLT2.OFFTRIMMODE = 1, then this register will supply offset trim when the ADC is in 12-bit single-ended mode for odd channels. Range is +127 steps to -128 steps (2's compliment format). Regardless of the converter resolution, the size of each trim step is (VREFHI-VREFLO)/65536. Reset type: XRSn
7-0	OFFTRIM	R/W	0h	ADC Offset Trim. Adjusts the conversion results of the converter up or down to account for offset error in the ADC. A different offset trim is required for each combination of resolution and signal mode. If ADCCTL2.OFFTRIMMODE = 0, then using the AdcSetMode function to set the resolution and signal mode will ensure that the correct offset trim is loaded into this register. If ADCCLT2.OFFTRIMMODE = 1, then this register will supply offset trim only when the ADC is in 12-bit single-ended mode and only for even channels. Range is +127 steps to -128 steps (2's compliment format). Regardless of the converter resolution, the size of each trim step is (VREFHI-VREFLO)/65536. Reset type: XRSn

24.16.3.57 ADCOFFTRIM2 Register (Offset = CCh) [Reset = 0000h]

ADCOFFTRIM2 is shown in Figure 24-160 and described in Table 24-138.

Return to the Summary Table.

ADC Offset Trim Register 2

Figure 24-160. ADCOFFTRIM2 Register

15	14	13	12	11	10	9	8
OFFTRIM16BSEODD							
R/W-0h							
7	6	5	4	3	2	1	0
OFFTRIM16BSEEVEN							
R/W-0h							

Table 24-138. ADCOFFTRIM2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	OFFTRIM16BSEODD	R/W	0h	If ADCCLT2.OFFTRIMMODE = 1, then this register will supply offset trim when the ADC is in 16-bit single-ended mode for odd channels. Range is +127 steps to -128 steps (2's compliment format). Regardless of the converter resolution, the size of each trim step is (VREFHI-VREFLO)/65536. Reset type: XRSn
7-0	OFFTRIM16BSEEVEN	R/W	0h	If ADCCLT2.OFFTRIMMODE = 1, then this register will supply offset trim when the ADC is in 16-bit single-ended mode for even channels. Range is +127 steps to -128 steps (2's compliment format). Regardless of the converter resolution, the size of each trim step is (VREFHI-VREFLO)/65536. Reset type: XRSn

24.16.3.58 ADCOFFTRIM3 Register (Offset = CEh) [Reset = 0000h]

ADCOFFTRIM3 is shown in Figure 24-161 and described in Table 24-139.

Return to the Summary Table.

ADC Offset Trim Register 3

Figure 24-161. ADCOFFTRIM3 Register

15	14	13	12	11	10	9	8
OFFTRIM16BDE							
R/W-0h							
7	6	5	4	3	2	1	0
OFFTRIM12BDE							
R/W-0h							

Table 24-139. ADCOFFTRIM3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	OFFTRIM16BDE	R/W	0h	If ADCCLT2.OFFTRIMMODE = 1, then this register will supply offset trim when the ADC is in 16-bit differential mode. Range is +127 steps to -128 steps (2's compliment format). Regardless of the converter resolution, the size of each trim step is (VREFHI-VREFLO)/65536. Reset type: XRSn
7-0	OFFTRIM12BDE	R/W	0h	If ADCCLT2.OFFTRIMMODE = 1, then this register will supply offset trim when the ADC is in 12-bit differential mode. Range is +127 steps to -128 steps (2's compliment format). Regardless of the converter resolution, the size of each trim step is (VREFHI-VREFLO)/65536. Reset type: XRSn

24.16.3.59 ADCPPB1CONFIG Register (Offset = D4h) [Reset = 0000h]

ADCPPB1CONFIG is shown in Figure 24-162 and described in Table 24-140.

Return to the Summary Table.

ADC PPB{#} Config Register

Figure 24-162. ADCPPB1CONFIG Register

15	14	13	12	11	10	9	8
RESERVED							DELTAEN
R-0h							R/W-0h
7	6	5	4	3	2	1	0
TWOSCOMPEN	ABSEN	CBCEN	CONFIG				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				

Table 24-140. ADCPPB1CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8	DELTAEN	R/W	0h	ADC Post Processing Block 1 enable delta (difference) from last sample calcualtion. When set, the ADCPPB1RESULT register will contain the difference between the most recent conversion result and the last value that would have been loaded into the ADCPPB1RESULT (if the delta calculation wasn't enabled). The delta calculation occurs after OFFREF, TWOSCOMPEN, and ABSEN calculations are applied. 0 Delta calculation disabled: no modification to ADCPPB1RESULT 1 ADCPPB1RESULT = ADCPPB1RESULT'[t] - ADCPPB1RESULT'[t - 1] Where ADCPPB1RESULT' is the value that would have been loaded into ADCPPB1RESULT without delta calculation Reset type: SYSRSn
7	TWOSCOMPEN	R/W	0h	ADC Post Processing Block 1 Two's Complement Enable. When set this bit enables the post conversion hardware processing circuit that performs a two's complement on the output of the offset/reference subtraction unit before storing the result in the ADCPPB1RESULT register. 0 ADCPPB1RESULT = ADCRESULTx - ADCPPB1OFFREF 1 ADCPPB1RESULT = ADCPPB1OFFREF - ADCRESULTx Reset type: SYSRSn
6	ABSEN	R/W	0h	ADC Post Processing Block 1 Absolute Value Enable. When set this bit enables absolute value calculation on the ADCRESULx associated with ADCPPB1. This occurs before the TWOSCOMPEN logic is evaluated (so enabling both TWOSCOMPEN and ABSEN will always result in a negative value stored in ADCPPBxRESULT) 0 ADCPPB1RESULT = ADCRESULTx - ADCPPB1OFFREF 1 ADCPPB1RESULT = abs(ADCRESULTx - ADCPPB1OFFREF) Reset type: SYSRSn
5	CBCEN	R/W	0h	ADC Post Processing Block Cycle By Cycle Enable. When set, this bit enables the post conversion hardware processing circuit to automatically clear the ADCEVTSTAT on a conversion if the event condition is no longer present. Reset type: SYSRSn

Table 24-140. ADCPPB1CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	CONFIG	R/W	0h	ADC Post Processing Block 1 Configuration. This bit field defines which SOC/EOC/RESULT is assocatied with this post processing block. 0x0 SOC0/EOC0/RESULT0 is associated with post processing block 1 0x1 SOC1/EOC1/RESULT1 is associated with post processing block 1 0x2 SOC2/EOC2/RESULT2 is associated with post processing block 1 0x3 SOC3/EOC3/RESULT3 is associated with post processing block 1 0x4 SOC4/EOC4/RESULT4 is associated with post processing block 1 0x5 SOC5/EOC5/RESULT5 is associated with post processing block 1 0x6 SOC6/EOC6/RESULT6 is associated with post processing block 1 0x7 SOC7/EOC7/RESULT7 is associated with post processing block 1 0x8 SOC8/EOC8/RESULT8 is associated with post processing block 1 0x9 SOC9/EOC9/RESULT9 is associated with post processing block 1 0xA SOC10/EOC10/RESULT10 is associated with post processing block 1 0xB SOC11/EOC11/RESULT11 is associated with post processing block 1 0xC SOC12/EOC12/RESULT12 is associated with post processing block 1 0xD SOC13/EOC13/RESULT13 is associated with post processing block 1 0xE SOC14/EOC14/RESULT14 is associated with post processing block 1 0xF SOC15/EOC15/RESULT15 is associated with post processing block 1 0x0 SOC16/EOC16/RESULT16 is associated with post processing block 1 0x1 SOC17/EOC17/RESULT17 is associated with post processing block 1 0x2 SOC18/EOC18/RESULT18 is associated with post processing block 1 0x3 SOC19/EOC19/RESULT19 is associated with post processing block 1 0x4 SOC20/EOC20/RESULT20 is associated with post processing block 1 0x5 SOC21/EOC21/RESULT21 is associated with post processing block 1 0x6 SOC22/EOC22/RESULT22 is associated with post processing block 1 0x7 SOC23/EOC23/RESULT23 is associated with post processing block 1 0x8 SOC24/EOC24/RESULT24 is associated with post processing block 1 0x9 SOC25/EOC25/RESULT25 is associated with post processing block 1 0xA SOC26/EOC26/RESULT26 is associated with post processing block 1 0xB SOC27/EOC27/RESULT27 is associated with post processing block 1 0xC SOC28/EOC28/RESULT28 is associated with post processing block 1 0xD SOC29/EOC29/RESULT29 is associated with post processing block 1 0xE SOC30/EOC30/RESULT30 is associated with post processing block 1

Table 24-140. ADCPPB1CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
				0xF SOC31/EOC31/RESULT31 is associated with post processing block 1 Reset type: SYSRSn

24.16.3.60 ADCPPB1STAMP Register (Offset = D6h) [Reset = 0000h]

ADCPPB1STAMP is shown in Figure 24-163 and described in Table 24-141.

Return to the Summary Table.

ADC PPB1 Sample Delay Time Stamp Register

Figure 24-163. ADCPPB1STAMP Register

15	14	13	12	11	10	9	8
RESERVED				DLYSTAMP			
R-0h				R-0h			
7	6	5	4	3	2	1	0
DLYSTAMP							
R-0h							

Table 24-141. ADCPPB1STAMP Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DLYSTAMP	R	0h	ADC Post Processing Block 1 Delay Time Stamp. When an SOC starts sampling the value contained in REQSTAMP is subtracted from the value in ADCCOUNTER.FREECOUNT and loaded into this bit field, thereby giving the number of system clock cycles delay between the SOC trigger and the actual start of the sample. Reset type: SYSRSn

24.16.3.61 ADCPPB1OFFCAL Register (Offset = D8h) [Reset = 0000h]

ADCPPB1OFFCAL is shown in Figure 24-164 and described in Table 24-142.

Return to the Summary Table.

ADC PPB1 Offset Calibration Register

Figure 24-164. ADCPPB1OFFCAL Register

15	14	13	12	11	10	9	8
RESERVED						OFFCAL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
OFFCAL							
R/W-0h							

Table 24-142. ADCPPB1OFFCAL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	OFFCAL	R/W	0h	ADC Post Processing Block 1 Offset Correction. This bit field can be used to digitally remove any system level offset inherent in the ADCIN circuit. This 10-bit signed value is subtracted from the ADC output before being stored in the ADCRESULT register. 000h No change. The ADC output is stored directly into ADCRESULT. 001h ADC output - 1 is stored into ADCRESULT. 002h ADC output - 2 is stored into ADCRESULT. ... 200h ADC output + 512 is stored into ADCRESULT. ... 3FFh ADC output + 1 is stored into ADCRESULT. NOTE: In 16-bit mode, the subtraction will saturate at 0000h and FFFFh before being stored into the ADCRESULT register. In 12-bit mode, the subtraction will saturate at 0000h and 0FFFh before being stored into the ADCRESULT register. Note: in the case that multiple PPBs point to the same SOC, only the OFFCAL of the lowest numbered PPB will be applied. Reset type: SYSRSn

24.16.3.62 ADCPPB1OFFREF Register (Offset = DAh) [Reset = 0000h]

ADCPPB1OFFREF is shown in Figure 24-165 and described in Table 24-143.

Return to the Summary Table.

ADC PPB1 Offset Reference Register

Figure 24-165. ADCPPB1OFFREF Register

15	14	13	12	11	10	9	8
OFFREF							
R/W-0h							
7	6	5	4	3	2	1	0
OFFREF							
R/W-0h							

Table 24-143. ADCPPB1OFFREF Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	OFFREF	R/W	0h	ADC Post Processing Block 1 Offset Correction. This bit field can be used to either calculate the feedback error or convert a unipolar signal to bipolar by subtracting a reference value. This 16-bit unsigned value is subtracted from the ADCRESULT register before being passed through an optional two's complement function and stored in the ADCPPB1RESULT register. This subtraction is not saturated. 0000h No change. The ADCRESULT value is passed on. 0001h ADCRESULT - 1 is passed on. 0002h ADCRESULT - 2 is passed on. ... 8000h ADCRESULT - 32,768 is passed on. ... FFFFh ADCRESULT - 65,535 is passed on. NOTE: In 12-bit mode the size of this register does not change from 16-bits. It is the user's responsibility to ensure that only a 12-bit value is written to this register when in 12-bit mode. Reset type: SYSRSn

24.16.3.63 ADCPPB1TRIPHI Register (Offset = DCh) [Reset = 00000000h]

ADCPPB1TRIPHI is shown in Figure 24-166 and described in Table 24-144.

Return to the Summary Table.

ADC PPB1 Trip High Register

Figure 24-166. ADCPPB1TRIPHI Register

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

Table 24-144. ADCPPB1TRIPHI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	LIMITHI	R/W	0h	ADC Post Processing Block 1 Trip High Limit. This value sets the digital comparator trip high limit. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPHI[23:17] will be ignored in 16 bit mode - TRIPHI[23:13] will be ignored in 12 bit mode Reset type: SYSRSn

24.16.3.64 ADCPPB1TRIPLO Register (Offset = E0h) [Reset = 00000000h]

ADCPPB1TRIPLO is shown in Figure 24-167 and described in Table 24-145.

Return to the Summary Table.

ADC PPB1 Trip Low/Trigger Time Stamp Register

Figure 24-167. ADCPPB1TRIPLO Register

31	30	29	28	27	26	25	24
REQSTAMP							
R-0h							
23	22	21	20	19	18	17	16
REQSTAMP				LIMITLO2EN	RESERVED		LSIGN
R-0h				R/W-0h	R-0h		R/W-0h
15	14	13	12	11	10	9	8
LIMITLO							
R/W-0h							
7	6	5	4	3	2	1	0
LIMITLO							
R/W-0h							

Table 24-145. ADCPPB1TRIPLO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	REQSTAMP	R	0h	ADC Post Processing Block 1 Request Time Stamp. When a trigger sets the associated SOC flag in the ADCSOCFLG1 register the value of ADCCOUNTER.FREECOUNT is loaded into this bit field. Reset type: SYSRSn
19	LIMITLO2EN	R/W	0h	Extended Low Limit 2 Enable. 0 = Low limit set by ADCPPB1TRIPLO register. Not compatible with comparison with ADCPPB1PSUM or ADCPPB1SUM 1 = Low limit set by ADCPPB1TRIPLO2 register Reset type: SYSRSn
18-17	RESERVED	R	0h	Reserved
16	LSIGN	R/W	0h	Low Limit Sign Bit. This is the sign bit (17th bit) to the LIMITLO bit field when in 16-bit ADC mode. Reset type: SYSRSn
15-0	LIMITLO	R/W	0h	ADC Post Processing Block 1 Trip Low Limit. This value sets the digital comparator trip low limit. In 16-bit mode all 17 bits will be compared against the 17 bits of the PPBRESULT bit field of the ADCPPB1RESULT register. In 12-bit mode bits 12:0 will be compared against bits 12:0 of the PPBRSULT bit field of the ADCPPB1RESULT register. Reset type: SYSRSn

24.16.3.65 ADCPPBTRIP1FILCTL Register (Offset = E4h) [Reset = 0000h]

ADCPPBTRIP1FILCTL is shown in Figure 24-168 and described in Table 24-146.

Return to the Summary Table.

ADCEVT1 Trip High Filter Control Register

Figure 24-168. ADCPPBTRIP1FILCTL Register

15	14	13	12	11	10	9	8
FILINIT	THRESH						SAMPWIN
R-0/W1S-0h	R/W-0h						R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN					RESERVED	FILTLOEN	FILTHIEN
R/W-0h					R-0h	R/W-0h	R/W-0h

Table 24-146. ADCPPBTRIP1FILCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	Trip filter initialization for PPB1. 0 No effect 1 Initialize all samples to the filter input value This applies to the filter on both the high and low trips. Reset type: SYSRSn
14-9	THRESH	R/W	0h	Trip filter majority voting threshold on PPB1. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Threshold used is THRESH+1. This applies to the filter on both the high and low trips. Reset type: SYSRSn
8-3	SAMPWIN	R/W	0h	Trip filter sample window size on PPB1. Number of samples to monitor is SAMPWIN+1. This applies to the filter on both the high and low trips. Reset type: SYSRSn
2	RESERVED	R	0h	Reserved
1	FILTLOEN	R/W	0h	ADC PPB1 TRIPLO Filter Enable 0 No filtering of PPB 1 trip low limit events 1 PPB1 trip high limit event filtering enabled Reset type: SYSRSn
0	FILTHIEN	R/W	0h	ADC PPB1 TRIPHI Filter Enable 0 No filtering of PPB 1 trip high limit events 1 PPB1 trip high limit event filtering enabled Reset type: SYSRSn

24.16.3.66 ADCPPBTRIP1FILCLKCTL Register (Offset = E8h) [Reset = 00000000h]

ADCPPBTRIP1FILCLKCTL is shown in Figure 24-169 and described in Table 24-147.

Return to the Summary Table.

ADCEVT1 Trip High Filter Prescale Control Register

Figure 24-169. ADCPPBTRIP1FILCLKCTL Register

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

Table 24-147. ADCPPBTRIP1FILCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	CLKPRESCALE	R/W	0h	ADCPPB1 filter sample clock prescale. The effective prescale value is (CLKPRESCALE + 1). This applies to the filter on both the high and low trips. Reset type: SYSRSn

24.16.3.67 ADCPPB2CONFIG Register (Offset = F4h) [Reset = 0001h]

ADCPPB2CONFIG is shown in Figure 24-170 and described in Table 24-148.

Return to the Summary Table.

ADC PPB{#} Config Register

Figure 24-170. ADCPPB2CONFIG Register

15	14	13	12	11	10	9	8
RESERVED							DELTAEN
R-0h							R/W-0h
7	6	5	4	3	2	1	0
TWOSCOMPEN	ABSEN	CBCEN	CONFIG				
R/W-0h	R/W-0h	R/W-0h	R/W-1h				

Table 24-148. ADCPPB2CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8	DELTAEN	R/W	0h	ADC Post Processing Block 2 enable delta (difference) from last sample calcualtion. When set, the ADCPPB2RESULT register will contain the difference between the most recent conversion result and the last value that would have been loaded into the ADCPPB2RESULT (if the delta calculation wasn't enabled). The delta calculation occurs after OFFREF, TWOSCOMPEN, and ABSEN calculations are applied. 0 Delta calculation disabled: no modification to ADCPPB2RESULT 1 ADCPPB2RESULT = ADCPPB2RESULT'[t] - ADCPPB2RESULT'[t - 1] Where ADCPPB2RESULT' is the value that would have been loaded into ADCPPB2RESULT without delta calculation Reset type: SYSRSn
7	TWOSCOMPEN	R/W	0h	ADC Post Processing Block 2 Two's Complement Enable. When set this bit enables the post conversion hardware processing circuit that performs a two's complement on the output of the offset/reference subtraction unit before storing the result in the ADCPPB2RESULT register. 0 ADCPPB2RESULT = ADCRESULTx - ADCPPB2OFFREF 1 ADCPPB2RESULT = ADCPPB2OFFREF - ADCRESULTx Reset type: SYSRSn
6	ABSEN	R/W	0h	ADC Post Processing Block 2 Absolute Value Enable. When set this bit enables absolute value calculation on the ADCRESULx associated with ADCPPB2. This occurs before the TWOSCOMPEN logic is evaluated (so enabling both TWOSCOMPEN and ABSEN will always result in a negative value stored in ADCPPBxRESULT) 0 ADCPPB2RESULT = ADCRESULTx - ADCPPB2OFFREF 1 ADCPPB2RESULT = abs(ADCRESULTx - ADCPPB2OFFREF) Reset type: SYSRSn
5	CBCEN	R/W	0h	ADC Post Processing Block Cycle By Cycle Enable. When set, this bit enables the post conversion hardware processing circuit to automatically clear the ADCEVTSTAT on a conversion if the event condition is no longer present. Reset type: SYSRSn

Table 24-148. ADCPPB2CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	CONFIG	R/W	1h	ADC Post Processing Block 2 Configuration. This bit field defines which SOC/EOC/RESULT is assocatied with this post processing block. 0x0 SOC0/EOC0/RESULT0 is associated with post processing block 2 0x1 SOC1/EOC1/RESULT1 is associated with post processing block 2 0x2 SOC2/EOC2/RESULT2 is associated with post processing block 2 0x3 SOC3/EOC3/RESULT3 is associated with post processing block 2 0x4 SOC4/EOC4/RESULT4 is associated with post processing block 2 0x5 SOC5/EOC5/RESULT5 is associated with post processing block 2 0x6 SOC6/EOC6/RESULT6 is associated with post processing block 2 0x7 SOC7/EOC7/RESULT7 is associated with post processing block 2 0x8 SOC8/EOC8/RESULT8 is associated with post processing block 2 0x9 SOC9/EOC9/RESULT9 is associated with post processing block 2 0xA SOC10/EOC10/RESULT10 is associated with post processing block 2 0xB SOC11/EOC11/RESULT11 is associated with post processing block 2 0xC SOC12/EOC12/RESULT12 is associated with post processing block 2 0xD SOC13/EOC13/RESULT13 is associated with post processing block 2 0xE SOC14/EOC14/RESULT14 is associated with post processing block 2 0xF SOC15/EOC15/RESULT15 is associated with post processing block 2 0x0 SOC16/EOC16/RESULT16 is associated with post processing block 2 0x1 SOC17/EOC17/RESULT17 is associated with post processing block 2 0x2 SOC18/EOC18/RESULT18 is associated with post processing block 2 0x3 SOC19/EOC19/RESULT19 is associated with post processing block 2 0x4 SOC20/EOC20/RESULT20 is associated with post processing block 2 0x5 SOC21/EOC21/RESULT21 is associated with post processing block 2 0x6 SOC22/EOC22/RESULT22 is associated with post processing block 2 0x7 SOC23/EOC23/RESULT23 is associated with post processing block 2 0x8 SOC24/EOC24/RESULT24 is associated with post processing block 2 0x9 SOC25/EOC25/RESULT25 is associated with post processing block 2 0xA SOC26/EOC26/RESULT26 is associated with post processing block 2 0xB SOC27/EOC27/RESULT27 is associated with post processing block 2 0xC SOC28/EOC28/RESULT28 is associated with post processing block 2 0xD SOC29/EOC29/RESULT29 is associated with post processing block 2 0xE SOC30/EOC30/RESULT30 is associated with post processing block 2

Table 24-148. ADCPPB2CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
				0xF SOC31/EOC31/RESULT31 is associated with post processing block 2 Reset type: SYSRSn

24.16.3.68 ADCPPB2STAMP Register (Offset = F6h) [Reset = 0000h]

ADCPPB2STAMP is shown in Figure 24-171 and described in Table 24-149.

Return to the Summary Table.

ADC PPB2 Sample Delay Time Stamp Register

Figure 24-171. ADCPPB2STAMP Register

15	14	13	12	11	10	9	8
RESERVED				DLYSTAMP			
R-0h				R-0h			
7	6	5	4	3	2	1	0
DLYSTAMP							
R-0h							

Table 24-149. ADCPPB2STAMP Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DLYSTAMP	R	0h	ADC Post Processing Block 2 Delay Time Stamp. When an SOC starts sampling the value contained in REQSTAMP is subtracted from the value in ADCCOUNTER.FREECOUNT and loaded into this bit field, thereby giving the number of system clock cycles delay between the SOC trigger and the actual start of the sample. Reset type: SYSRSn

24.16.3.69 ADCPPB2OFFCAL Register (Offset = F8h) [Reset = 0000h]

ADCPPB2OFFCAL is shown in Figure 24-172 and described in Table 24-150.

Return to the Summary Table.

ADC PPB2 Offset Calibration Register

Figure 24-172. ADCPPB2OFFCAL Register

15	14	13	12	11	10	9	8
RESERVED						OFFCAL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
OFFCAL							
R/W-0h							

Table 24-150. ADCPPB2OFFCAL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	OFFCAL	R/W	0h	ADC Post Processing Block 2 Offset Correction. This bit field can be used to digitally remove any system level offset inherent in the ADCIN circuit. This 10-bit signed value is subtracted from the ADC output before being stored in the ADCRESULT register. 000h No change. The ADC output is stored directly into ADCRESULT. 001h ADC output - 1 is stored into ADCRESULT. 002h ADC output - 2 is stored into ADCRESULT. ... 200h ADC output + 512 is stored into ADCRESULT. ... 3FFh ADC output + 1 is stored into ADCRESULT. NOTE: In 16-bit mode, the subtraction will saturate at 0000h and FFFFh before being stored into the ADCRESULT register. In 12-bit mode, the subtraction will saturate at 0000h and 0FFFh before being stored into the ADCRESULT register. Note: in the case that multiple PPBs point to the same SOC, only the OFFCAL of the lowest numbered PPB will be applied. Reset type: SYSRSn

24.16.3.70 ADCPPB2OFFREF Register (Offset = FAh) [Reset = 0000h]

ADCPPB2OFFREF is shown in Figure 24-173 and described in Table 24-151.

Return to the Summary Table.

ADC PPB2 Offset Reference Register

Figure 24-173. ADCPPB2OFFREF Register

15	14	13	12	11	10	9	8
OFFREF							
R/W-0h							
7	6	5	4	3	2	1	0
OFFREF							
R/W-0h							

Table 24-151. ADCPPB2OFFREF Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	OFFREF	R/W	0h	ADC Post Processing Block 2 Offset Correction. This bit field can be used to either calculate the feedback error or convert a unipolar signal to bipolar by subtracting a reference value. This 16-bit unsigned value is subtracted from the ADCRESULT register before being passed through an optional two's complement function and stored in the ADCPPB2RESULT register. This subtraction is not saturated. 0000h No change. The ADCRESULT value is passed on. 0001h ADCRESULT - 1 is passed on. 0002h ADCRESULT - 2 is passed on. ... 8000h ADCRESULT - 32,768 is passed on. ... FFFFh ADCRESULT - 65,535 is passed on. NOTE: In 12-bit mode the size of this register does not change from 16-bits. It is the user's responsibility to ensure that only a 12-bit value is written to this register when in 12-bit mode. Reset type: SYSRSn

24.16.3.71 ADCPPB2TRIPHI Register (Offset = FCh) [Reset = 00000000h]

ADCPPB2TRIPHI is shown in Figure 24-174 and described in Table 24-152.

Return to the Summary Table.

ADC PPB2 Trip High Register

Figure 24-174. ADCPPB2TRIPHI Register

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

Table 24-152. ADCPPB2TRIPHI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	LIMITHI	R/W	0h	ADC Post Processing Block 2 Trip High Limit. This value sets the digital comparator trip high limit. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPHI[23:17] will be ignored in 16 bit mode - TRIPHI[23:13] will be ignored in 12 bit mode Reset type: SYSRSn

24.16.3.72 ADCPPB2TRIPLO Register (Offset = 100h) [Reset = 00000000h]

ADCPPB2TRIPLO is shown in Figure 24-175 and described in Table 24-153.

Return to the Summary Table.

ADC PPB2 Trip Low/Trigger Time Stamp Register

Figure 24-175. ADCPPB2TRIPLO Register

31	30	29	28	27	26	25	24
REQSTAMP							
R-0h							
23	22	21	20	19	18	17	16
REQSTAMP				LIMITLO2EN	RESERVED		LSIGN
R-0h				R/W-0h	R-0h		R/W-0h
15	14	13	12	11	10	9	8
LIMITLO							
R/W-0h							
7	6	5	4	3	2	1	0
LIMITLO							
R/W-0h							

Table 24-153. ADCPPB2TRIPLO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	REQSTAMP	R	0h	ADC Post Processing Block 2 Request Time Stamp. When a trigger sets the associated SOC flag in the ADCSOCFLG1 register the value of ADCCOUNTER.FREECOUNT is loaded into this bit field. Reset type: SYSRSn
19	LIMITLO2EN	R/W	0h	Extended Low Limit 2 Enable. 0 = Low limit set by ADCPPB2TRIPLO register. Not compatible with comparison with ADCPPB2PSUM or ADCPPB2SUM 1 = Low limit set by ADCPPB2TRIPLO2 register Reset type: SYSRSn
18-17	RESERVED	R	0h	Reserved
16	LSIGN	R/W	0h	Low Limit Sign Bit. This is the sign bit (17th bit) to the LIMITLO bit field when in 16-bit ADC mode. Reset type: SYSRSn
15-0	LIMITLO	R/W	0h	ADC Post Processing Block 2 Trip Low Limit. This value sets the digital comparator trip low limit. In 16-bit mode all 17 bits will be compared against the 17 bits of the PPBRESULT bit field of the ADCPPB2RESULT register. In 12-bit mode bits 12:0 will be compared against bits 12:0 of the PPBRSULT bit field of the ADCPPB2RESULT register. Reset type: SYSRSn

24.16.3.73 ADCPPBTRIP2FILCTL Register (Offset = 104h) [Reset = 0000h]

ADCPPBTRIP2FILCTL is shown in Figure 24-176 and described in Table 24-154.

Return to the Summary Table.

ADCEVT2 Trip High Filter Control Register

Figure 24-176. ADCPPBTRIP2FILCTL Register

15	14	13	12	11	10	9	8
FILINIT	THRESH						SAMPWIN
R-0/W1S-0h	R/W-0h						R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN					RESERVED	FILTLOEN	FILTHIEN
R/W-0h					R-0h	R/W-0h	R/W-0h

Table 24-154. ADCPPBTRIP2FILCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	Trip filter initialization for PPB2. 0 No effect 1 Initialize all samples to the filter input value This applies to the filter on both the high and low trips. Reset type: SYSRSn
14-9	THRESH	R/W	0h	Trip filter majority voting threshold on PPB2. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Threshold used is THRESH+1. This applies to the filter on both the high and low trips. Reset type: SYSRSn
8-3	SAMPWIN	R/W	0h	Trip filter sample window size on PPB2. Number of samples to monitor is SAMPWIN+1. This applies to the filter on both the high and low trips. Reset type: SYSRSn
2	RESERVED	R	0h	Reserved
1	FILTLOEN	R/W	0h	ADC PPB2 TRIPLO Filter Enable 0 No filtering of PPB 2 trip low limit events 1 PPB2 trip high limit event filtering enabled Reset type: SYSRSn
0	FILTHIEN	R/W	0h	ADC PPB2 TRIPHI Filter Enable 0 No filtering of PPB 2 trip high limit events 1 PPB2 trip high limit event filtering enabled Reset type: SYSRSn

24.16.3.74 ADCPPBTRIP2FILCLKCTL Register (Offset = 108h) [Reset = 00000000h]

ADCPPBTRIP2FILCLKCTL is shown in Figure 24-177 and described in Table 24-155.

Return to the Summary Table.

ADCEVT2 Trip High Filter Prescale Control Register

Figure 24-177. ADCPPBTRIP2FILCLKCTL Register

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

Table 24-155. ADCPPBTRIP2FILCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	CLKPRESCALE	R/W	0h	ADCPPB2 filter sample clock prescale. The effective prescale value is (CLKPRESCALE + 1). This applies to the filter on both the high and low trips. Reset type: SYSRSn

24.16.3.75 ADCPPB3CONFIG Register (Offset = 114h) [Reset = 0002h]

ADCPPB3CONFIG is shown in Figure 24-178 and described in Table 24-156.

Return to the Summary Table.

ADC PPB{#} Config Register

Figure 24-178. ADCPPB3CONFIG Register

15	14	13	12	11	10	9	8
RESERVED							DELTAEN
R-0h							R/W-0h
7	6	5	4	3	2	1	0
TWOSCOMPEN	ABSEN	CBCEN	CONFIG				
R/W-0h	R/W-0h	R/W-0h	R/W-2h				

Table 24-156. ADCPPB3CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8	DELTAEN	R/W	0h	ADC Post Processing Block 3 enable delta (difference) from last sample calcualtion. When set, the ADCPPB3RESULT register will contain the difference between the most recent conversion result and the last value that would have been loaded into the ADCPPB3RESULT (if the delta calculation wasn't enabled). The delta calculation occurs after OFFREF, TWOSCOMPEN, and ABSEN calculations are applied. 0 Delta calculation disabled: no modification to ADCPPB3RESULT 1 ADCPPB3RESULT = ADCPPB3RESULT'[t] - ADCPPB3RESULT'[t - 1] Where ADCPPB3RESULT' is the value that would have been loaded into ADCPPB3RESULT without delta calculation Reset type: SYSRSn
7	TWOSCOMPEN	R/W	0h	ADC Post Processing Block 3 Two's Complement Enable. When set this bit enables the post conversion hardware processing circuit that performs a two's complement on the output of the offset/reference subtraction unit before storing the result in the ADCPPB3RESULT register. 0 ADCPPB3RESULT = ADCRESULTx - ADCPPB3OFFREF 1 ADCPPB3RESULT = ADCPPB3OFFREF - ADCRESULTx Reset type: SYSRSn
6	ABSEN	R/W	0h	ADC Post Processing Block 3 Absolute Value Enable. When set this bit enables absolute value calculation on the ADCRESULx associated with ADCPPB3. This occurs before the TWOSCOMPEN logic is evaluated (so enabling both TWOSCOMPEN and ABSEN will always result in a negative value stored in ADCPPBxRESULT) 0 ADCPPB3RESULT = ADCRESULTx - ADCPPB3OFFREF 1 ADCPPB3RESULT = abs(ADCRESULTx - ADCPPB3OFFREF) Reset type: SYSRSn
5	CBCEN	R/W	0h	ADC Post Processing Block Cycle By Cycle Enable. When set, this bit enables the post conversion hardware processing circuit to automatically clear the ADCEVTSTAT on a conversion if the event condition is no longer present. Reset type: SYSRSn

Table 24-156. ADCPPB3CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	CONFIG	R/W	2h	ADC Post Processing Block 3 Configuration. This bit field defines which SOC/EOC/RESULT is assocatied with this post processing block. 0x0 SOC0/EOC0/RESULT0 is associated with post processing block 3 0x1 SOC1/EOC1/RESULT1 is associated with post processing block 3 0x2 SOC2/EOC2/RESULT2 is associated with post processing block 3 0x3 SOC3/EOC3/RESULT3 is associated with post processing block 3 0x4 SOC4/EOC4/RESULT4 is associated with post processing block 3 0x5 SOC5/EOC5/RESULT5 is associated with post processing block 3 0x6 SOC6/EOC6/RESULT6 is associated with post processing block 3 0x7 SOC7/EOC7/RESULT7 is associated with post processing block 3 0x8 SOC8/EOC8/RESULT8 is associated with post processing block 3 0x9 SOC9/EOC9/RESULT9 is associated with post processing block 3 0xA SOC10/EOC10/RESULT10 is associated with post processing block 3 0xB SOC11/EOC11/RESULT11 is associated with post processing block 3 0xC SOC12/EOC12/RESULT12 is associated with post processing block 3 0xD SOC13/EOC13/RESULT13 is associated with post processing block 3 0xE SOC14/EOC14/RESULT14 is associated with post processing block 3 0xF SOC15/EOC15/RESULT15 is associated with post processing block 3 0x0 SOC16/EOC16/RESULT16 is associated with post processing block 3 0x1 SOC17/EOC17/RESULT17 is associated with post processing block 3 0x2 SOC18/EOC18/RESULT18 is associated with post processing block 3 0x3 SOC19/EOC19/RESULT19 is associated with post processing block 3 0x4 SOC20/EOC20/RESULT20 is associated with post processing block 3 0x5 SOC21/EOC21/RESULT21 is associated with post processing block 3 0x6 SOC22/EOC22/RESULT22 is associated with post processing block 3 0x7 SOC23/EOC23/RESULT23 is associated with post processing block 3 0x8 SOC24/EOC24/RESULT24 is associated with post processing block 3 0x9 SOC25/EOC25/RESULT25 is associated with post processing block 3 0xA SOC26/EOC26/RESULT26 is associated with post processing block 3 0xB SOC27/EOC27/RESULT27 is associated with post processing block 3 0xC SOC28/EOC28/RESULT28 is associated with post processing block 3 0xD SOC29/EOC29/RESULT29 is associated with post processing block 3 0xE SOC30/EOC30/RESULT30 is associated with post processing block 3

Table 24-156. ADCPPB3CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
				0xF SOC31/EOC31/RESULT31 is associated with post processing block 3 Reset type: SYSRSn

24.16.3.76 ADCPPB3STAMP Register (Offset = 116h) [Reset = 0000h]

ADCPPB3STAMP is shown in Figure 24-179 and described in Table 24-157.

Return to the Summary Table.

ADC PPB3 Sample Delay Time Stamp Register

Figure 24-179. ADCPPB3STAMP Register

15	14	13	12	11	10	9	8
RESERVED				DLYSTAMP			
R-0h				R-0h			
7	6	5	4	3	2	1	0
DLYSTAMP							
R-0h							

Table 24-157. ADCPPB3STAMP Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DLYSTAMP	R	0h	ADC Post Processing Block 3 Delay Time Stamp. When an SOC starts sampling the value contained in REQSTAMP is subtracted from the value in ADCCOUNTER.FREECOUNT and loaded into this bit field, thereby giving the number of system clock cycles delay between the SOC trigger and the actual start of the sample. Reset type: SYSRSn

24.16.3.77 ADCPPB3OFFCAL Register (Offset = 118h) [Reset = 0000h]

ADCPPB3OFFCAL is shown in Figure 24-180 and described in Table 24-158.

Return to the Summary Table.

ADC PPB3 Offset Calibration Register

Figure 24-180. ADCPPB3OFFCAL Register

15	14	13	12	11	10	9	8
RESERVED						OFFCAL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
OFFCAL							
R/W-0h							

Table 24-158. ADCPPB3OFFCAL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	OFFCAL	R/W	0h	ADC Post Processing Block 3 Offset Correction. This bit field can be used to digitally remove any system level offset inherent in the ADCIN circuit. This 10-bit signed value is subtracted from the ADC output before being stored in the ADCRESULT register. 000h No change. The ADC output is stored directly into ADCRESULT. 001h ADC output - 1 is stored into ADCRESULT. 002h ADC output - 2 is stored into ADCRESULT. ... 200h ADC output + 512 is stored into ADCRESULT. ... 3FFh ADC output + 1 is stored into ADCRESULT. NOTE: In 16-bit mode, the subtraction will saturate at 0000h and FFFFh before being stored into the ADCRESULT register. In 12-bit mode, the subtraction will saturate at 0000h and 0FFFh before being stored into the ADCRESULT register. Note: in the case that multiple PPBs point to the same SOC, only the OFFCAL of the lowest numbered PPB will be applied. Reset type: SYSRSn

24.16.3.78 ADCPPB3OFFREF Register (Offset = 11Ah) [Reset = 0000h]

ADCPPB3OFFREF is shown in Figure 24-181 and described in Table 24-159.

Return to the Summary Table.

ADC PPB3 Offset Reference Register

Figure 24-181. ADCPPB3OFFREF Register

15	14	13	12	11	10	9	8
OFFREF							
R/W-0h							
7	6	5	4	3	2	1	0
OFFREF							
R/W-0h							

Table 24-159. ADCPPB3OFFREF Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	OFFREF	R/W	0h	ADC Post Processing Block 3 Offset Correction. This bit field can be used to either calculate the feedback error or convert a unipolar signal to bipolar by subtracting a reference value. This 16-bit unsigned value is subtracted from the ADCRESULT register before being passed through an optional two's complement function and stored in the ADCPPB3RESULT register. This subtraction is not saturated. 0000h No change. The ADCRESULT value is passed on. 0001h ADCRESULT - 1 is passed on. 0002h ADCRESULT - 2 is passed on. ... 8000h ADCRESULT - 32,768 is passed on. ... FFFFh ADCRESULT - 65,535 is passed on. NOTE: In 12-bit mode the size of this register does not change from 16-bits. It is the user's responsibility to ensure that only a 12-bit value is written to this register when in 12-bit mode. Reset type: SYSRSn

24.16.3.79 ADCPPB3TRIPHI Register (Offset = 11Ch) [Reset = 00000000h]

ADCPPB3TRIPHI is shown in Figure 24-182 and described in Table 24-160.

Return to the Summary Table.

ADC PPB3 Trip High Register

Figure 24-182. ADCPPB3TRIPHI Register

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

Table 24-160. ADCPPB3TRIPHI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	LIMITHI	R/W	0h	ADC Post Processing Block 3 Trip High Limit. This value sets the digital comparator trip high limit. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPHI[23:17] will be ignored in 16 bit mode - TRIPHI[23:13] will be ignored in 12 bit mode Reset type: SYSRSn

24.16.3.80 ADCPPB3TRIPLO Register (Offset = 120h) [Reset = 00000000h]

ADCPPB3TRIPLO is shown in Figure 24-183 and described in Table 24-161.

Return to the Summary Table.

ADC PPB3 Trip Low/Trigger Time Stamp Register

Figure 24-183. ADCPPB3TRIPLO Register

31	30	29	28	27	26	25	24
REQSTAMP							
R-0h							
23	22	21	20	19	18	17	16
REQSTAMP				LIMITLO2EN	RESERVED		LSIGN
R-0h				R/W-0h	R-0h		R/W-0h
15	14	13	12	11	10	9	8
LIMITLO							
R/W-0h							
7	6	5	4	3	2	1	0
LIMITLO							
R/W-0h							

Table 24-161. ADCPPB3TRIPLO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	REQSTAMP	R	0h	ADC Post Processing Block 3 Request Time Stamp. When a trigger sets the associated SOC flag in the ADCSOCFLG1 register the value of ADCCOUNTER.FREECOUNT is loaded into this bit field. Reset type: SYSRSn
19	LIMITLO2EN	R/W	0h	Extended Low Limit 2 Enable. 0 = Low limit set by ADCPPB3TRIPLO register. Not compatible with comparison with ADCPPB3PSUM or ADCPPB3SUM 1 = Low limit set by ADCPPB3TRIPLO2 register Reset type: SYSRSn
18-17	RESERVED	R	0h	Reserved
16	LSIGN	R/W	0h	Low Limit Sign Bit. This is the sign bit (17th bit) to the LIMITLO bit field when in 16-bit ADC mode. Reset type: SYSRSn
15-0	LIMITLO	R/W	0h	ADC Post Processing Block 3 Trip Low Limit. This value sets the digital comparator trip low limit. In 16-bit mode all 17 bits will be compared against the 17 bits of the PPBRESULT bit field of the ADCPPB3RESULT register. In 12-bit mode bits 12:0 will be compared against bits 12:0 of the PPBRSULT bit field of the ADCPPB3RESULT register. Reset type: SYSRSn

24.16.3.81 ADCPPBTRIP3FILCTL Register (Offset = 124h) [Reset = 0000h]

ADCPPBTRIP3FILCTL is shown in Figure 24-184 and described in Table 24-162.

Return to the Summary Table.

ADCEVT3 Trip High Filter Control Register

Figure 24-184. ADCPPBTRIP3FILCTL Register

15	14	13	12	11	10	9	8
FILINIT	THRESH						SAMPWIN
R-0/W1S-0h	R/W-0h						R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN					RESERVED	FILTLOEN	FILTHIEN
R/W-0h					R-0h	R/W-0h	R/W-0h

Table 24-162. ADCPPBTRIP3FILCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	Trip filter initialization for PPB3. 0 No effect 1 Initialize all samples to the filter input value This applies to the filter on both the high and low trips. Reset type: SYSRSn
14-9	THRESH	R/W	0h	Trip filter majority voting threshold on PPB3. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Threshold used is THRESH+1. This applies to the filter on both the high and low trips. Reset type: SYSRSn
8-3	SAMPWIN	R/W	0h	Trip filter sample window size on PPB3. Number of samples to monitor is SAMPWIN+1. This applies to the filter on both the high and low trips. Reset type: SYSRSn
2	RESERVED	R	0h	Reserved
1	FILTLOEN	R/W	0h	ADC PPB3 TRIPLO Filter Enable 0 No filtering of PPB 3 trip low limit events 1 PPB3 trip high limit event filtering enabled Reset type: SYSRSn
0	FILTHIEN	R/W	0h	ADC PPB3 TRIPHI Filter Enable 0 No filtering of PPB 3 trip high limit events 1 PPB3 trip high limit event filtering enabled Reset type: SYSRSn

24.16.3.82 ADCPPBTRIP3FILCLKCTL Register (Offset = 128h) [Reset = 00000000h]

ADCPPBTRIP3FILCLKCTL is shown in Figure 24-185 and described in Table 24-163.

Return to the Summary Table.

ADCEVT3 Trip High Filter Prescale Control Register

Figure 24-185. ADCPPBTRIP3FILCLKCTL Register

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

Table 24-163. ADCPPBTRIP3FILCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	CLKPRESCALE	R/W	0h	ADCPPB3 filter sample clock prescale. The effective prescale value is (CLKPRESCALE + 1). This applies to the filter on both the high and low trips. Reset type: SYSRSn

24.16.3.83 ADCPPB4CONFIG Register (Offset = 134h) [Reset = 0003h]

ADCPPB4CONFIG is shown in Figure 24-186 and described in Table 24-164.

Return to the Summary Table.

ADC PPB{#} Config Register

Figure 24-186. ADCPPB4CONFIG Register

15	14	13	12	11	10	9	8
RESERVED							DELTAEN
R-0h							R/W-0h
7	6	5	4	3	2	1	0
TWOSCOMPEN	ABSEN	CBCEN	CONFIG				
R/W-0h	R/W-0h	R/W-0h	R/W-3h				

Table 24-164. ADCPPB4CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8	DELTAEN	R/W	0h	ADC Post Processing Block 4 enable delta (difference) from last sample calcualtion. When set, the ADCPPB4RESULT register will contain the difference between the most recent conversion result and the last value that would have been loaded into the ADCPPB4RESULT (if the delta calculation wasn't enabled). The delta calculation occurs after OFFREF, TWOSCOMPEN, and ABSEN calculations are applied. 0 Delta calculation disabled: no modification to ADCPPB4RESULT 1 ADCPPB4RESULT = ADCPPB4RESULT'[t] - ADCPPB4RESULT'[t - 1] Where ADCPPB4RESULT' is the value that would have been loaded into ADCPPB4RESULT without delta calculation Reset type: SYSRSn
7	TWOSCOMPEN	R/W	0h	ADC Post Processing Block 4 Two's Complement Enable. When set this bit enables the post conversion hardware processing circuit that performs a two's complement on the output of the offset/reference subtraction unit before storing the result in the ADCPPB4RESULT register. 0 ADCPPB4RESULT = ADCRESULTx - ADCPPB4OFFREF 1 ADCPPB4RESULT = ADCPPB4OFFREF - ADCRESULTx Reset type: SYSRSn
6	ABSEN	R/W	0h	ADC Post Processing Block 4 Absolute Value Enable. When set this bit enables absolute value calculation on the ADCRESULx associated with ADCPPB4. This occurs before the TWOSCOMPEN logic is evaluated (so enabling both TWOSCOMPEN and ABSEN will always result in a negative value stored in ADCPPBxRESULT) 0 ADCPPB4RESULT = ADCRESULTx - ADCPPB4OFFREF 1 ADCPPB4RESULT = abs(ADCRESULTx - ADCPPB4OFFREF) Reset type: SYSRSn
5	CBCEN	R/W	0h	ADC Post Processing Block Cycle By Cycle Enable. When set, this bit enables the post conversion hardware processing circuit to automatically clear the ADCEVTSTAT on a conversion if the event condition is no longer present. Reset type: SYSRSn

Table 24-164. ADCPPB4CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	CONFIG	R/W	3h	ADC Post Processing Block 4 Configuration. This bit field defines which SOC/EOC/RESULT is assocatied with this post processing block. 0x0 SOC0/EOC0/RESULT0 is associated with post processing block 4 0x1 SOC1/EOC1/RESULT1 is associated with post processing block 4 0x2 SOC2/EOC2/RESULT2 is associated with post processing block 4 0x3 SOC3/EOC3/RESULT3 is associated with post processing block 4 0x4 SOC4/EOC4/RESULT4 is associated with post processing block 4 0x5 SOC5/EOC5/RESULT5 is associated with post processing block 4 0x6 SOC6/EOC6/RESULT6 is associated with post processing block 4 0x7 SOC7/EOC7/RESULT7 is associated with post processing block 4 0x8 SOC8/EOC8/RESULT8 is associated with post processing block 4 0x9 SOC9/EOC9/RESULT9 is associated with post processing block 4 0xA SOC10/EOC10/RESULT10 is associated with post processing block 4 0xB SOC11/EOC11/RESULT11 is associated with post processing block 4 0xC SOC12/EOC12/RESULT12 is associated with post processing block 4 0xD SOC13/EOC13/RESULT13 is associated with post processing block 4 0xE SOC14/EOC14/RESULT14 is associated with post processing block 4 0xF SOC15/EOC15/RESULT15 is associated with post processing block 4 0x0 SOC16/EOC16/RESULT16 is associated with post processing block 4 0x1 SOC17/EOC17/RESULT17 is associated with post processing block 4 0x2 SOC18/EOC18/RESULT18 is associated with post processing block 4 0x3 SOC19/EOC19/RESULT19 is associated with post processing block 4 0x4 SOC20/EOC20/RESULT20 is associated with post processing block 4 0x5 SOC21/EOC21/RESULT21 is associated with post processing block 4 0x6 SOC22/EOC22/RESULT22 is associated with post processing block 4 0x7 SOC23/EOC23/RESULT23 is associated with post processing block 4 0x8 SOC24/EOC24/RESULT24 is associated with post processing block 4 0x9 SOC25/EOC25/RESULT25 is associated with post processing block 4 0xA SOC26/EOC26/RESULT26 is associated with post processing block 4 0xB SOC27/EOC27/RESULT27 is associated with post processing block 4 0xC SOC28/EOC28/RESULT28 is associated with post processing block 4 0xD SOC29/EOC29/RESULT29 is associated with post processing block 4 0xE SOC30/EOC30/RESULT30 is associated with post processing block 4

Table 24-164. ADCPPB4CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
				0xF SOC31/EOC31/RESULT31 is associated with post processing block 4 Reset type: SYSRSn

24.16.3.84 ADCPPB4STAMP Register (Offset = 136h) [Reset = 0000h]

ADCPPB4STAMP is shown in Figure 24-187 and described in Table 24-165.

Return to the Summary Table.

ADC PPB4 Sample Delay Time Stamp Register

Figure 24-187. ADCPPB4STAMP Register

15	14	13	12	11	10	9	8
RESERVED				DLYSTAMP			
R-0h				R-0h			
7	6	5	4	3	2	1	0
DLYSTAMP							
R-0h							

Table 24-165. ADCPPB4STAMP Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DLYSTAMP	R	0h	ADC Post Processing Block 4 Delay Time Stamp. When an SOC starts sampling the value contained in REQSTAMP is subtracted from the value in ADCCOUNTER.FREECOUNT and loaded into this bit field, thereby giving the number of system clock cycles delay between the SOC trigger and the actual start of the sample. Reset type: SYSRSn

24.16.3.85 ADCPPB4OFFCAL Register (Offset = 138h) [Reset = 0000h]

ADCPPB4OFFCAL is shown in Figure 24-188 and described in Table 24-166.

Return to the Summary Table.

ADC PPB4 Offset Calibration Register

Figure 24-188. ADCPPB4OFFCAL Register

15	14	13	12	11	10	9	8
RESERVED						OFFCAL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
OFFCAL							
R/W-0h							

Table 24-166. ADCPPB4OFFCAL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	OFFCAL	R/W	0h	ADC Post Processing Block 4 Offset Correction. This bit field can be used to digitally remove any system level offset inherent in the ADCIN circuit. This 10-bit signed value is subtracted from the ADC output before being stored in the ADCRESULT register. 000h No change. The ADC output is stored directly into ADCRESULT. 001h ADC output - 1 is stored into ADCRESULT. 002h ADC output - 2 is stored into ADCRESULT. ... 200h ADC output + 512 is stored into ADCRESULT. ... 3FFh ADC output + 1 is stored into ADCRESULT. NOTE: In 16-bit mode, the subtraction will saturate at 0000h and FFFFh before being stored into the ADCRESULT register. In 12-bit mode, the subtraction will saturate at 0000h and 0FFFh before being stored into the ADCRESULT register. Note: in the case that multiple PPBs point to the same SOC, only the OFFCAL of the lowest numbered PPB will be applied. Reset type: SYSRSn

24.16.3.86 ADCPPB4OFFREF Register (Offset = 13Ah) [Reset = 0000h]

ADCPPB4OFFREF is shown in Figure 24-189 and described in Table 24-167.

Return to the Summary Table.

ADC PPB4 Offset Reference Register

Figure 24-189. ADCPPB4OFFREF Register

15	14	13	12	11	10	9	8
OFFREF							
R/W-0h							
7	6	5	4	3	2	1	0
OFFREF							
R/W-0h							

Table 24-167. ADCPPB4OFFREF Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	OFFREF	R/W	0h	ADC Post Processing Block 4 Offset Correction. This bit field can be used to either calculate the feedback error or convert a unipolar signal to bipolar by subtracting a reference value. This 16-bit unsigned value is subtracted from the ADCRESULT register before being passed through an optional two's complement function and stored in the ADCPPB4RESULT register. This subtraction is not saturated. 0000h No change. The ADCRESULT value is passed on. 0001h ADCRESULT - 1 is passed on. 0002h ADCRESULT - 2 is passed on. ... 8000h ADCRESULT - 32,768 is passed on. ... FFFFh ADCRESULT - 65,535 is passed on. NOTE: In 12-bit mode the size of this register does not change from 16-bits. It is the user's responsibility to ensure that only a 12-bit value is written to this register when in 12-bit mode. Reset type: SYSRSn

24.16.3.87 ADCPPB4TRIPHI Register (Offset = 13Ch) [Reset = 00000000h]

ADCPPB4TRIPHI is shown in Figure 24-190 and described in Table 24-168.

Return to the Summary Table.

ADC PPB4 Trip High Register

Figure 24-190. ADCPPB4TRIPHI Register

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

Table 24-168. ADCPPB4TRIPHI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	LIMITHI	R/W	0h	ADC Post Processing Block 4 Trip High Limit. This value sets the digital comparator trip high limit. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPHI[23:17] will be ignored in 16 bit mode - TRIPHI[23:13] will be ignored in 12 bit mode Reset type: SYSRSn

24.16.3.88 ADCPPB4TRIPLO Register (Offset = 140h) [Reset = 00000000h]

ADCPPB4TRIPLO is shown in Figure 24-191 and described in Table 24-169.

Return to the Summary Table.

ADC PPB4 Trip Low/Trigger Time Stamp Register

Figure 24-191. ADCPPB4TRIPLO Register

31	30	29	28	27	26	25	24
REQSTAMP							
R-0h							
23	22	21	20	19	18	17	16
REQSTAMP				LIMITLO2EN	RESERVED		LSIGN
R-0h				R/W-0h	R-0h		R/W-0h
15	14	13	12	11	10	9	8
LIMITLO							
R/W-0h							
7	6	5	4	3	2	1	0
LIMITLO							
R/W-0h							

Table 24-169. ADCPPB4TRIPLO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	REQSTAMP	R	0h	ADC Post Processing Block 4 Request Time Stamp. When a trigger sets the associated SOC flag in the ADCSOCFLG1 register the value of ADCCOUNTER.FREECOUNT is loaded into this bit field. Reset type: SYSRSn
19	LIMITLO2EN	R/W	0h	Extended Low Limit 2 Enable. 0 = Low limit set by ADCPPB4TRIPLO register. Not compatible with comparison with ADCPPB4PSUM or ADCPPB4SUM 1 = Low limit set by ADCPPB4TRIPLO2 register Reset type: SYSRSn
18-17	RESERVED	R	0h	Reserved
16	LSIGN	R/W	0h	Low Limit Sign Bit. This is the sign bit (17th bit) to the LIMITLO bit field when in 16-bit ADC mode. Reset type: SYSRSn
15-0	LIMITLO	R/W	0h	ADC Post Processing Block 4 Trip Low Limit. This value sets the digital comparator trip low limit. In 16-bit mode all 17 bits will be compared against the 17 bits of the PPBRESULT bit field of the ADCPPB4RESULT register. In 12-bit mode bits 12:0 will be compared against bits 12:0 of the PPBRSULT bit field of the ADCPPB4RESULT register. Reset type: SYSRSn

24.16.3.89 ADCPPBTRIP4FILCTL Register (Offset = 144h) [Reset = 0000h]

ADCPPBTRIP4FILCTL is shown in Figure 24-192 and described in Table 24-170.

Return to the Summary Table.

ADCEVT4 Trip High Filter Control Register

Figure 24-192. ADCPPBTRIP4FILCTL Register

15	14	13	12	11	10	9	8
FILINIT	THRESH						SAMPWIN
R-0/W1S-0h	R/W-0h						R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN					RESERVED	FILTLOEN	FILTHIEN
R/W-0h					R-0h	R/W-0h	R/W-0h

Table 24-170. ADCPPBTRIP4FILCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	Trip filter initialization for PPB4. 0 No effect 1 Initialize all samples to the filter input value This applies to the filter on both the high and low trips. Reset type: SYSRSn
14-9	THRESH	R/W	0h	Trip filter majority voting threshold on PPB4. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Threshold used is THRESH+1. This applies to the filter on both the high and low trips. Reset type: SYSRSn
8-3	SAMPWIN	R/W	0h	Trip filter sample window size on PPB4. Number of samples to monitor is SAMPWIN+1. This applies to the filter on both the high and low trips. Reset type: SYSRSn
2	RESERVED	R	0h	Reserved
1	FILTLOEN	R/W	0h	ADC PPB4 TRIPLO Filter Enable 0 No filtering of PPB 4 trip low limit events 1 PPB4 trip high limit event filtering enabled Reset type: SYSRSn
0	FILTHIEN	R/W	0h	ADC PPB4 TRIPHI Filter Enable 0 No filtering of PPB 4 trip high limit events 1 PPB4 trip high limit event filtering enabled Reset type: SYSRSn

24.16.3.90 ADCPPBTRIP4FILCLKCTL Register (Offset = 148h) [Reset = 00000000h]

ADCPPBTRIP4FILCLKCTL is shown in Figure 24-193 and described in Table 24-171.

Return to the Summary Table.

ADCEVT4 Trip High Filter Prescale Control Register

Figure 24-193. ADCPPBTRIP4FILCLKCTL Register

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

Table 24-171. ADCPPBTRIP4FILCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	CLKPRESCALE	R/W	0h	ADCPPB4 filter sample clock prescale. The effective prescale value is (CLKPRESCALE + 1). This applies to the filter on both the high and low trips. Reset type: SYSRSn

24.16.3.91 ADCSAFECHECKRESEN Register (Offset = 154h) [Reset = 00000000h]

ADCSAFECHECKRESEN is shown in Figure 24-194 and described in Table 24-172.

Return to the Summary Table.

ADC Safe Check Result Enable Register

Figure 24-194. ADCSAFECHECKRESEN Register

31	30	29	28	27	26	25	24
SOC15CHKEN		SOC14CHKEN		SOC13CHKEN		SOC12CHKEN	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
SOC11CHKEN		SOC10CHKEN		SOC9CHKEN		SOC8CHKEN	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
SOC7CHKEN		SOC6CHKEN		SOC5CHKEN		SOC4CHKEN	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
SOC3CHKEN		SOC2CHKEN		SOC1CHKEN		SOC0CHKEN	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 24-172. ADCSAFECHECKRESEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SOC15CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT15 passed to safety checker 10 PPB Result associated with SOC15 passed to safety checker 11 PPB Sum associated with SOC15 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
29-28	SOC14CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT14 passed to safety checker 10 PPB Result associated with SOC14 passed to safety checker 11 PPB Sum associated with SOC14 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
27-26	SOC13CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT13 passed to safety checker 10 PPB Result associated with SOC13 passed to safety checker 11 PPB Sum associated with SOC13 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn

Table 24-172. ADCSAFECHECKRESEN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25-24	SOC12CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT12 passed to safety checker 10 PPB Result associated with SOC12 passed to safety checker 11 PPB Sum associated with SOC12 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
23-22	SOC11CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT11 passed to safety checker 10 PPB Result associated with SOC11 passed to safety checker 11 PPB Sum associated with SOC11 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
21-20	SOC10CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT10 passed to safety checker 10 PPB Result associated with SOC10 passed to safety checker 11 PPB Sum associated with SOC10 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
19-18	SOC9CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT9 passed to safety checker 10 PPB Result associated with SOC9 passed to safety checker 11 PPB Sum associated with SOC9 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
17-16	SOC8CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT8 passed to safety checker 10 PPB Result associated with SOC8 passed to safety checker 11 PPB Sum associated with SOC8 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn

Table 24-172. ADCSAFECHECKRESEN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-14	SOC7CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT7 passed to safety checker 10 PPB Result associated with SOC7 passed to safety checker 11 PPB Sum associated with SOC7 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
13-12	SOC6CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT6 passed to safety checker 10 PPB Result associated with SOC6 passed to safety checker 11 PPB Sum associated with SOC6 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
11-10	SOC5CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT5 passed to safety checker 10 PPB Result associated with SOC5 passed to safety checker 11 PPB Sum associated with SOC5 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
9-8	SOC4CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT4 passed to safety checker 10 PPB Result associated with SOC4 passed to safety checker 11 PPB Sum associated with SOC4 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
7-6	SOC3CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT3 passed to safety checker 10 PPB Result associated with SOC3 passed to safety checker 11 PPB Sum associated with SOC3 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn

Table 24-172. ADCSAFECHECKRESEN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	SOC2CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT2 passed to safety checker 10 PPB Result associated with SOC2 passed to safety checker 11 PPB Sum associated with SOC2 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
3-2	SOC1CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT1 passed to safety checker 10 PPB Result associated with SOC1 passed to safety checker 11 PPB Sum associated with SOC1 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
1-0	SOC0CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT0 passed to safety checker 10 PPB Result associated with SOC0 passed to safety checker 11 PPB Sum associated with SOC0 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn

24.16.3.92 ADCSAFECHECKRESEN2 Register (Offset = 158h) [Reset = 00000000h]

ADCSAFECHECKRESEN2 is shown in Figure 24-195 and described in Table 24-173.

Return to the Summary Table.

ADC Safe Check Result Enable 2 Register

Figure 24-195. ADCSAFECHECKRESEN2 Register

31	30	29	28	27	26	25	24
SOC31CHKEN		SOC30CHKEN		SOC29CHKEN		SOC28CHKEN	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
SOC27CHKEN		SOC26CHKEN		SOC25CHKEN		SOC24CHKEN	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
SOC23CHKEN		SOC22CHKEN		SOC21CHKEN		SOC20CHKEN	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
SOC19CHKEN		SOC18CHKEN		SOC17CHKEN		SOC16CHKEN	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 24-173. ADCSAFECHECKRESEN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SOC31CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT31 passed to safety checker 10 PPB Result associated with SOC31 passed to safety checker 11 PPB Sum associated with SOC31 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
29-28	SOC30CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT30 passed to safety checker 10 PPB Result associated with SOC30 passed to safety checker 11 PPB Sum associated with SOC30 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
27-26	SOC29CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT29 passed to safety checker 10 PPB Result associated with SOC29 passed to safety checker 11 PPB Sum associated with SOC29 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn

Table 24-173. ADCSAFECHECKRESEN2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25-24	SOC28CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT28 passed to safety checker 10 PPB Result associated with SOC28 passed to safety checker 11 PPB Sum associated with SOC28 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
23-22	SOC27CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT27 passed to safety checker 10 PPB Result associated with SOC27 passed to safety checker 11 PPB Sum associated with SOC27 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
21-20	SOC26CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT26 passed to safety checker 10 PPB Result associated with SOC26 passed to safety checker 11 PPB Sum associated with SOC26 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
19-18	SOC25CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT25 passed to safety checker 10 PPB Result associated with SOC25 passed to safety checker 11 PPB Sum associated with SOC25 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
17-16	SOC24CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT24 passed to safety checker 10 PPB Result associated with SOC24 passed to safety checker 11 PPB Sum associated with SOC24 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn

Table 24-173. ADCSAFECHECKRESEN2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-14	SOC23CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT23 passed to safety checker 10 PPB Result associated with SOC23 passed to safety checker 11 PPB Sum associated with SOC23 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
13-12	SOC22CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT22 passed to safety checker 10 PPB Result associated with SOC22 passed to safety checker 11 PPB Sum associated with SOC22 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
11-10	SOC21CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT21 passed to safety checker 10 PPB Result associated with SOC21 passed to safety checker 11 PPB Sum associated with SOC21 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
9-8	SOC20CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT20 passed to safety checker 10 PPB Result associated with SOC20 passed to safety checker 11 PPB Sum associated with SOC20 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
7-6	SOC19CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT19 passed to safety checker 10 PPB Result associated with SOC19 passed to safety checker 11 PPB Sum associated with SOC19 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn

Table 24-173. ADCSAFECHECKRESEN2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	SOC18CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT18 passed to safety checker 10 PPB Result associated with SOC18 passed to safety checker 11 PPB Sum associated with SOC18 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
3-2	SOC17CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT17 passed to safety checker 10 PPB Result associated with SOC17 passed to safety checker 11 PPB Sum associated with SOC17 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn
1-0	SOC16CHKEN	R/W	0h	Determine which result will be passed to the safety result checker. Only one of the raw ADC result, the PPB result, or the final PPB accumulated SUM can be passed on to the checker for each conversion. 00 No result passed to safety checker 01 ADCRESULT16 passed to safety checker 10 PPB Result associated with SOC16 passed to safety checker 11 PPB Sum associated with SOC16 passed to safety chekcer Note: if multiple PPBs point to the same SOC, the lowest numbered PPB will have priority to pass its result to the safety checker Reset type: SYSRSn

24.16.3.93 ADCINTCYCLE Register (Offset = 172h) [Reset = 0000h]

ADCINTCYCLE is shown in Figure 24-196 and described in Table 24-174.

Return to the Summary Table.

ADC Early Interrupt Generation Cycle

Figure 24-196. ADCINTCYCLE Register

15	14	13	12	11	10	9	8
DELAY							
R/W-0h							
7	6	5	4	3	2	1	0
DELAY							
R/W-0h							

Table 24-174. ADCINTCYCLE Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	DELAY	R/W	0h	ADC Early Interrupt Generation Cycle Delay: Defines the delay from the fall edge of ADCSOC in terms of system clock cycles, for the interrupt to be generated. Reset type: SYSRSn

24.16.3.94 ADCINLTRIM1 Register (Offset = 174h) [Reset = X0000000h]

ADCINLTRIM1 is shown in Figure 24-197 and described in Table 24-175.

Return to the Summary Table.

ADC Linearity Trim 1 Register

Figure 24-197. ADCINLTRIM1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INLTRIM31TO0																															
R/W-Xh																															

Table 24-175. ADCINLTRIM1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INLTRIM31TO0	R/W	Xh	ADC Linearity Trim Bits 31-0. This register should not be modified unless specifically indicated by TI Errata or other documentation. Modifying the contents of this register could cause this module to operate outside of datasheet specifications. Reset type: XRSn

24.16.3.95 ADCINLTRIM2 Register (Offset = 178h) [Reset = X0000000h]

ADCINLTRIM2 is shown in Figure 24-198 and described in Table 24-176.

Return to the Summary Table.

ADC Linearity Trim 2 Register

Figure 24-198. ADCINLTRIM2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INLTRIM63TO32																															
R/W-Xh																															

Table 24-176. ADCINLTRIM2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INLTRIM63TO32	R/W	Xh	ADC Linearity Trim Bits 63-32. This register should not be modified unless specifically indicated by TI Errata or other documentation. Modifying the contents of this register could cause this module to operate outside of datasheet specifications. Reset type: XRSn

24.16.3.96 ADCINLTRIM3 Register (Offset = 17Ch) [Reset = X0000000h]

ADCINLTRIM3 is shown in Figure 24-199 and described in Table 24-177.

Return to the Summary Table.

ADC Linearity Trim 3 Register

Figure 24-199. ADCINLTRIM3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INLTRIM95TO64																															
R/W-Xh																															

Table 24-177. ADCINLTRIM3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INLTRIM95TO64	R/W	Xh	ADC Linearity Trim Bits 95-64. This register should not be modified unless specifically indicated by TI Errata or other documentation. Modifying the contents of this register could cause this module to operate outside of datasheet specifications. Reset type: XRSn

24.16.3.97 ADCINLTRIM4 Register (Offset = 180h) [Reset = X0000000h]

ADCINLTRIM4 is shown in Figure 24-200 and described in Table 24-178.

Return to the Summary Table.

ADC Linearity Trim 4 Register

Figure 24-200. ADCINLTRIM4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INLTRIM127TO96																															
R/W-Xh																															

Table 24-178. ADCINLTRIM4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INLTRIM127TO96	R/W	Xh	ADC Linearity Trim Bits 127-96. This register should not be modified unless specifically indicated by TI Errata or other documentation. Modifying the contents of this register could cause this module to operate outside of datasheet specifications. Reset type: XRSn

24.16.3.98 ADCINLTRIM5 Register (Offset = 184h) [Reset = X0000000h]

ADCINLTRIM5 is shown in Figure 24-201 and described in Table 24-179.

Return to the Summary Table.

ADC Linearity Trim 5 Register

Figure 24-201. ADCINLTRIM5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INLTRIM159TO128																															
R/W-Xh																															

Table 24-179. ADCINLTRIM5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INLTRIM159TO128	R/W	Xh	ADC Linearity Trim Bits 159-128. This register should not be modified unless specifically indicated by TI Errata or other documentation. Modifying the contents of this register could cause this module to operate outside of datasheet specifications. Reset type: XRSn

24.16.3.99 ADCINLTRIM6 Register (Offset = 188h) [Reset = X0000000h]

ADCINLTRIM6 is shown in Figure 24-202 and described in Table 24-180.

Return to the Summary Table.

ADC Linearity Trim 6 Register

Figure 24-202. ADCINLTRIM6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INLTRIM191TO160																															
R/W-Xh																															

Table 24-180. ADCINLTRIM6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INLTRIM191TO160	R/W	Xh	ADC Linearity Trim Bits 191-160. This register should not be modified unless specifically indicated by TI Errata or other documentation. Modifying the contents of this register could cause this module to operate outside of datasheet specifications. Reset type: XRSn

24.16.3.100 ADCREV2 Register (Offset = 18Eh) [Reset = 0205h]

ADCREV2 is shown in Figure 24-203 and described in Table 24-181.

Return to the Summary Table.

ADC Wrapper Revision Register

Figure 24-203. ADCREV2 Register

15	14	13	12	11	10	9	8
WRAPPERREV							
R-2h							
7	6	5	4	3	2	1	0
WRAPPERTYPE							
R-5h							

Table 24-181. ADCREV2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	WRAPPERREV	R	2h	ADC Revision. To allow documentation of differences between revisions. First version is labeled as 00h. 01h : Wrapper for MicroADC with 32 SOC 02h : Wrapper for C2KADC with 32 SOC Reset type: SYSRSn
7-0	WRAPPERTYPE	R	5h	ADC Wrapper Type. Always set to 5 for type 5 ADC Wrapper. Reset type: SYSRSn

24.16.3.101 REP1CTL Register (Offset = 194h) [Reset = 00000000h]

REP1CTL is shown in Figure 24-204 and described in Table 24-182.

Return to the Summary Table.

ADC Trigger Repeater 1 Control Register

Figure 24-204. REP1CTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
SWSYNC	RESERVED	SYNCINSEL					
R-0/W1S-0h	R-0h	R/W-0h					
15	14	13	12	11	10	9	8
RESERVED	TRIGGER						
R-0h	R/W-0h						
7	6	5	4	3	2	1	0
TRIGGEROVF	PHASEOVF	RESERVED	RESERVED	MODULEBUSY	RESERVED	ACTIVEMODE	MODE
R/W1C-0h	R/W1C-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R/W-0h

Table 24-182. REP1CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23	SWSYNC	R-0/W1S	0h	Trigger repeater 1 software force sync. On a sync. event, all registers in repeater 1 are reset to a ready and waiting state. Values of NSEL, PHASE, and MODE are preserved. Note: SOCs associated with repeater 1 are not cleared. Reset type: SYSRSn
22	RESERVED	R	0h	Reserved
21-16	SYNCINSEL	R/W	0h	Trigger repeater 1 sync. input select. On a sync. event, all registers in repeater 1 are reset to a ready and waiting state. Values of NSEL, PHASE, and MODE are preserved. Note: SOCs associated with repeater 1 are not cleared. Refer to SOC spec for more details Reset type: SYSRSn
15	RESERVED	R	0h	Reserved

Table 24-182. REP1CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14-8	TRIGGER	R/W	0h	ADC Trigger Repeater 1 Trigger Select. Selects the trigger to modify via oversampling or undersampling. 00h REPTRIG0 - Software only 01h REPTRIG1 - CPU1 Timer 0, TINT0n 02h REPTRIG2 - CPU1 Timer 1, TINT1n 03h REPTRIG3 - CPU1 Timer 2, TINT2n 04h REPTRIG4 - GPIO, Input X-Bar INPUT5 05h REPTRIG5 - ePWM1, ADCSOCA 06h REPTRIG6 - ePWM1, ADCSOCB 07h REPTRIG7 - ePWM2, ADCSOCA 08h REPTRIG8 - ePWM2, ADCSOCB 09h REPTRIG9 - ePWM3, ADCSOCA 0Ah REPTRIG10 - ePWM3, ADCSOCB 0Bh REPTRIG11 - ePWM4, ADCSOCA 0Ch REPTRIG12 - ePWM4, ADCSOCB 0Dh REPTRIG13 - ePWM5, ADCSOCA 0Eh REPTRIG14 - ePWM5, ADCSOCB 0Fh REPTRIG15 - ePWM6, ADCSOCA 10h REPTRIG16 - ePWM6, ADCSOCB 11h REPTRIG17 - ePWM7, ADCSOCA 12h REPTRIG18 - ePWM7, ADCSOCB 13h REPTRIG19 - ePWM8, ADCSOCA 14h REPTRIG20 - ePWM8, ADCSOCB 15h REPTRIG21 - ePWM9, ADCSOCA 16h REPTRIG22 - ePWM9, ADCSOCB 17h REPTRIG23 - ePWM10, ADCSOCA 18h REPTRIG24 - ePWM10, ADCSOCB 19h REPTRIG25 - ePWM11, ADCSOCA 1Ah REPTRIG26 - ePWM11, ADCSOCB 1Bh REPTRIG27 - ePWM12, ADCSOCA 1Ch REPTRIG28 - ePWM12, ADCSOCB 1Dh REPTRIG29 - CPU2 Timer 0, TINT0n 1Eh REPTRIG30 - CPU2 Timer 1, TINT1n 1Fh REPTRIG31 - CPU2 Timer 2, TINT2n 20h - 4Fh - Reserved 50h REPTRIG80 eCAP1 51h REPTRIG81 eCAP2 52h REPTRIG82 eCAP3 53h REPTRIG83 eCAP4 54h REPTRIG84 eCAP5 55h REPTRIG85 eCAP6 56h REPTRIG86 eCAP7 57h REPTRIG87 eCAP8 58h REPTRIG88 - ePWM13, ADCSOCA 59h REPTRIG89 - ePWM13, ADCSOCB 5Ah REPTRIG90 - ePWM14, ADCSOCA 5Bh REPTRIG91 - ePWM14, ADCSOCB 5Ch REPTRIG92 - ePWM15, ADCSOCA 5Dh REPTRIG93 - ePWM15, ADCSOCB 5Eh REPTRIG94 - ePWM16, ADCSOCA 5Fh REPTRIG95 - ePWM16, ADCSOCB 60h REPTRIG96 - ePWM17, ADCSOCA 61h REPTRIG97 - ePWM17, ADCSOCB 62h REPTRIG98 - ePWM18, ADCSOCA 63h REPTRIG99 - ePWM18, ADCSOCB 64h - 7Fh - Reserved Reset type: SYSRSn

Table 24-182. REP1CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	TRIGGEROVF	R/W1C	0h	ADC Trigger Repeater 1 Oversampled Trigger Overflow. Indicates that a trigger was dropped because a trigger arrived while the repeater was still generating repeated oversampled triggers (NCOUNT was not 0 or SOCs associated with Repeater 1 were still pending). Writing a 1 will clear this flag. Note: This flag won't be set in undersampling mode or when NSEL = 0 if a trigger arrives before the previous SOCs have completed, the trigger will be passed and the overflow flags of the SOCs that were still pending will be set. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
6	PHASEOVF	R/W1C	0h	ADC Trigger Repeater 1 Phase Delay Overflow. Indicates that a trigger was dropped because a trigger arrived when the phase delay logic was still waiting to send the delayed trigger (PHASECOUNT was not 0). Writing a 1 will clear this flag. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	MODULEBUSY	R	0h	ADC Trigger Repeater 1 Module Busy indicator. In oversampling mode: 0 = Repeater 1 is idle and can accept a new repeated trigger in oversampling mode 1 = Repeater 1 still has repeated triggers remaining (NCOUNT > 0) or associated SOCs are still pending (SOCBUSY is 1) If a new oversampled trigger is received while the module is still busy, the TRIGGEROVF bit will be set and the trigger will be ignored. Reset type: SYSRSn
2	RESERVED	R	0h	Reserved
1	ACTIVEMODE	R	0h	When a trigger is recived in oversampling or undersampling mode the value of MODE is copied to ACTIVEMODE. ACTIVEMODE determines if the repeater will repeat of filter triggers. Changes to MODE while the repeater is working therefore won't cause any changes in functionality until the module becomes idle and then a new trigger is received. 0 = module is oversampling 1 = module is undersampling Reset type: SYSRSn
0	MODE	R/W	0h	ADC trigger repeater 1 mode selection. Select either oversampling or undersampling mode. In oversampling mode, when the trigger selected by REP1CTL.TRIGSEL is received, the repeater will repeat the trigger REP1N.NSEL + 1 times. In undersampling mode, when the trigger selected by REP1CTL.TRIGSEL is received the first time, the repeater will pass the trigger through. The next REP1N.NSEL triggers will be ignored. 0 = oversampling 1 = undersampling Reset type: SYSRSn

24.16.3.102 REP1N Register (Offset = 198h) [Reset = 00000000h]

REP1N is shown in Figure 24-205 and described in Table 24-183.

Return to the Summary Table.

ADC Trigger Repeater 1 N Select Register

Figure 24-205. REP1N Register

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

Table 24-183. REP1N Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	0h	Reserved
22-16	NCOUNT	R	0h	ADC trigger repeater 1 trigger count. In oversampling mode, indicates the number of triggers remaining to be generated. If a trigger is received corresponding to REP1CTL.TRIGSEL while NCOUNT is not 0 (the repeater is still busy generating the repeated triggers) then the trigger will be ignored and REP1CTL.TRIGOVF will be set to 1. In undersampling mode, indicates the number of triggers remaining to be supressed. Reset type: SYSRSn
15-7	RESERVED	R	0h	Reserved
6-0	NSEL	R/W	0h	ADC Trigger Repeater 1 selection of number of triggers. In oversampling mode, selects the number of repeated triggers. For each trigger received corresponding to REP1CTL.TRIGSEL, NSEL + 1 triggers will be generated. 0 = 1 trigger is generated (pass-through) 1 = 2 triggers are generated 2 = 3 triggers are generated ... 127 = 128 triggers are generated In undersampling mode, selects the number triggers to be supressed. 1 out NSEL + 1 triggers received corresponding to REP1CTL.TRIGSEL will be passed through (the first trigger will be passed through and the subsequent NSEL triggers will be supressed). 0 = all triggers are passed 1 = 1 out of 2 triggers are passed 2 = 1 out of 3 triggers are passed ... 127 = 1 out of 128 triggers are passed Reset type: SYSRSn

24.16.3.103 REP1PHASE Register (Offset = 19Ch) [Reset = 00000000h]

REP1PHASE is shown in Figure 24-206 and described in Table 24-184.

Return to the Summary Table.

ADC Trigger Repeater 1 Phase Select Register

Figure 24-206. REP1PHASE Register

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

Table 24-184. REP1PHASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PHASECOUNT	R	0h	ADC trigger repeater 1 phase delay status. When the trigger selected by REP1CTL.TRIGSEL is received, this register will start counting down from PHASECOUNT until the counter reaches 0, at which point the trigger will be passed on to the repeater re-trigger logic. If the trigger selected by REP1CTL.TRIGSEL is recieved when PHASECOUNT is not 0 (the phase delay logic is busy from the previous trigger) then the new trigger will be ignored and REP1CTL.PHASEOVF will be set to 1. Reset type: SYSRSn
15-0	PHASE	R/W	0h	ADC trigger repeater 1 phase delay configuration. Defines the number of SYSCLKs to delay the selected trigger before passing it on to the re-triggering logic. 0 = trigger is passed through without delay 1 = trigger is delayed by 1 SYSCLK 2 = trigger is delayed by 2 SYSCLKs ... 65535 = trigger is delayed by 65535 SYSCLKs Reset type: SYSRSn

24.16.3.104 REP1SPREAD Register (Offset = 1A0h) [Reset = 00000000h]

REP1SPREAD is shown in Figure 24-207 and described in Table 24-185.

Return to the Summary Table.

ADC Trigger Repeater 1 Spread Select Register

Figure 24-207. REP1SPREAD Register

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

Table 24-185. REP1SPREAD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SPREADCOUNT	R	0h	ADC trigger repeater 1 spread status. When a trigger is sent to the ADC in oversampling mode, this register will start counting down from SPREAD until SPREADCOUNT equals 0. The next repeated trigger to the ADC in oversampling mode will not occur until SPREADCOUNT is 0 (minimum time is complete) and REP1CTL.BUSY = 0 (SOCs associated with trigger repeater 1 are no longer pending). Reset type: SYSRSn
15-0	SPREAD	R/W	0h	ADC trigger repeater 1 spread delay configuration. In oversampling mode, defines the minimum number of SYSCLKs to wait before creating the next repeated trigger to the ADC. If SPREAD is less than the time needed for all SOCs associated with repeater 1 to sample and convert, then the repeater will generate triggers as fast as the ADC can convert the associated conversions. If SPREAD is greater than the time needed for all SOCs associated with repeater 1 to sample and convert, then repeated triggers to the ADC will be SPREAD SYSCLK cycles apart. 0 = oversampled repeated triggers occur as fast as the ADC can sample and convert associated SOCs 1 = time between repeated triggers is at least 1 SYSCLKs 2 = time between repeated triggers is at least 2 SYSCLKs ... 65535 = time between repeated triggers is at least 65535 SYSCLKs Reset type: SYSRSn

24.16.3.105 REP1FRC Register (Offset = 1A4h) [Reset = 0000h]

REP1FRC is shown in Figure 24-208 and described in Table 24-186.

Return to the Summary Table.

ADC Trigger Repeater 1 Software Force Register

Figure 24-208. REP1FRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							SWFRC
R-0h							R-0/W1S-0h

Table 24-186. REP1FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R	0h	Reserved
0	SWFRC	R-0/W1S	0h	Write 1 to force a trigger to repeat block 1 input regardless of the value of TRIGGER. Always reads 0. Reset type: SYSRSn

24.16.3.106 REP2CTL Register (Offset = 1B4h) [Reset = 00000000h]

REP2CTL is shown in Figure 24-209 and described in Table 24-187.

Return to the Summary Table.

ADC Trigger Repeater 2 Control Register

Figure 24-209. REP2CTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
SWSYNC	RESERVED	SYNCINSEL					
R-0/W1S-0h	R-0h	R/W-0h					
15	14	13	12	11	10	9	8
RESERVED	TRIGGER						
R-0h	R/W-0h						
7	6	5	4	3	2	1	0
TRIGGEROVF	PHASEOVF	RESERVED	RESERVED	MODULEBUSY	RESERVED	ACTIVEMODE	MODE
R/W1C-0h	R/W1C-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R/W-0h

Table 24-187. REP2CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23	SWSYNC	R-0/W1S	0h	Trigger repeater 2 software force sync. On a sync. event, all registers in repeater 2 are reset to a ready and waiting state. Values of NSEL, PHASE, and MODE are preserved. Note: SOCs associated with repeater 2 are not cleared. Reset type: SYSRSn
22	RESERVED	R	0h	Reserved
21-16	SYNCINSEL	R/W	0h	Trigger repeater 2 sync. input select. On a sync. event, all registers in repeater 2 are reset to a ready and waiting state. Values of NSEL, PHASE, and MODE are preserved. Note: SOCs associated with repeater 2 are not cleared. Refer to SOC spec for more details Reset type: SYSRSn
15	RESERVED	R	0h	Reserved

Table 24-187. REP2CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14-8	TRIGGER	R/W	0h	ADC Trigger Repeater 2 Trigger Select. Selects the trigger to modify via oversampling or undersampling. 00h REPTRIG0 - Software only 01h REPTRIG1 - CPU1 Timer 0, TINT0n 02h REPTRIG2 - CPU1 Timer 1, TINT1n 03h REPTRIG3 - CPU1 Timer 2, TINT2n 04h REPTRIG4 - GPIO, Input X-Bar INPUT5 05h REPTRIG5 - ePWM1, ADCSOCA 06h REPTRIG6 - ePWM1, ADCSOCB 07h REPTRIG7 - ePWM2, ADCSOCA 08h REPTRIG8 - ePWM2, ADCSOCB 09h REPTRIG9 - ePWM3, ADCSOCA 0Ah REPTRIG10 - ePWM3, ADCSOCB 0Bh REPTRIG11 - ePWM4, ADCSOCA 0Ch REPTRIG12 - ePWM4, ADCSOCB 0Dh REPTRIG13 - ePWM5, ADCSOCA 0Eh REPTRIG14 - ePWM5, ADCSOCB 0Fh REPTRIG15 - ePWM6, ADCSOCA 10h REPTRIG16 - ePWM6, ADCSOCB 11h REPTRIG17 - ePWM7, ADCSOCA 12h REPTRIG18 - ePWM7, ADCSOCB 13h REPTRIG19 - ePWM8, ADCSOCA 14h REPTRIG20 - ePWM8, ADCSOCB 15h REPTRIG21 - ePWM9, ADCSOCA 16h REPTRIG22 - ePWM9, ADCSOCB 17h REPTRIG23 - ePWM10, ADCSOCA 18h REPTRIG24 - ePWM10, ADCSOCB 19h REPTRIG25 - ePWM11, ADCSOCA 1Ah REPTRIG26 - ePWM11, ADCSOCB 1Bh REPTRIG27 - ePWM12, ADCSOCA 1Ch REPTRIG28 - ePWM12, ADCSOCB 1Dh REPTRIG29 - CPU2 Timer 0, TINT0n 1Eh REPTRIG30 - CPU2 Timer 1, TINT1n 1Fh REPTRIG31 - CPU2 Timer 2, TINT2n 20h - 4Fh - Reserved 50h REPTRIG80 eCAP1 51h REPTRIG81 eCAP2 52h REPTRIG82 eCAP3 53h REPTRIG83 eCAP4 54h REPTRIG84 eCAP5 55h REPTRIG85 eCAP6 56h REPTRIG86 eCAP7 57h REPTRIG87 eCAP8 58h REPTRIG88 - ePWM13, ADCSOCA 59h REPTRIG89 - ePWM13, ADCSOCB 5Ah REPTRIG90 - ePWM14, ADCSOCA 5Bh REPTRIG91 - ePWM14, ADCSOCB 5Ch REPTRIG92 - ePWM15, ADCSOCA 5Dh REPTRIG93 - ePWM15, ADCSOCB 5Eh REPTRIG94 - ePWM16, ADCSOCA 5Fh REPTRIG95 - ePWM16, ADCSOCB 60h REPTRIG96 - ePWM17, ADCSOCA 61h REPTRIG97 - ePWM17, ADCSOCB 62h REPTRIG98 - ePWM18, ADCSOCA 63h REPTRIG99 - ePWM18, ADCSOCB 64h - 7Fh - Reserved Reset type: SYSRSn

Table 24-187. REP2CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	TRIGGEROVF	R/W1C	0h	ADC Trigger Repeater 2 Oversampled Trigger Overflow. Indicates that a trigger was dropped because a trigger arrived while the repeater was still generating repeated oversampled triggers (NCOUNT was not 0 or SOCs associated with Repeater 2 were still pending). Writing a 1 will clear this flag. Note: This flag won't be set in undersampling mode or when NSEL = 0 if a trigger arrives before the previous SOCs have completed, the trigger will be passed and the overflow flags of the SOCs that were still pending will be set. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
6	PHASEOVF	R/W1C	0h	ADC Trigger Repeater 2 Phase Delay Overflow. Indicates that a trigger was dropped because a trigger arrived when the phase delay logic was still waiting to send the delayed trigger (PHASECOUNT was not 0). Writing a 1 will clear this flag. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	MODULEBUSY	R	0h	ADC Trigger Repeater 2 Module Busy indicator. In oversampling mode: 0 = Repeater 2 is idle and can accept a new repeated trigger in oversampling mode 1 = Repeater 2 still has repeated triggers remaining (NCOUNT > 0) or associated SOCs are still pending (SOCBUSY is 1) If a new oversampled trigger is received while the module is still busy, the TRIGGEROVF bit will be set and the trigger will be ignored. Reset type: SYSRSn
2	RESERVED	R	0h	Reserved
1	ACTIVEMODE	R	0h	When a trigger is recived in oversampling or undersampling mode the value of MODE is copied to ACTIVEMODE. ACTIVEMODE determines if the repeater will repeat of filter triggers. Changes to MODE while the repeater is working therefore won't cause any changes in functionality until the module becomes idle and then a new trigger is received. 0 = module is oversampling 1 = module is undersampling Reset type: SYSRSn
0	MODE	R/W	0h	ADC trigger repeater 2 mode selection. Select either oversampling or undersampling mode. In oversampling mode, when the trigger selected by REP2CTL.TRIGSEL is received, the repeater will repeat the trigger REP2N.NSEL + 1 times. In undersampling mode, when the trigger selected by REP2CTL.TRIGSEL is received the first time, the repeater will pass the trigger through. The next REP2N.NSEL triggers will be ignored. 0 = oversampling 1 = undersampling Reset type: SYSRSn

24.16.3.107 REP2N Register (Offset = 1B8h) [Reset = 00000000h]

REP2N is shown in Figure 24-210 and described in Table 24-188.

Return to the Summary Table.

ADC Trigger Repeater 2 N Select Register

Figure 24-210. REP2N Register

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

Table 24-188. REP2N Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	0h	Reserved
22-16	NCOUNT	R	0h	ADC trigger repeater 2 trigger count. In oversampling mode, indicates the number of triggers remaining to be generated. If a trigger is received corresponding to REP2CTL.TRIGSEL while NCOUNT is not 0 (the repeater is still busy generating the repeated triggers) then the trigger will be ignored and REP2CTL.TRIGOVF will be set to 1. In undersampling mode, indicates the number of triggers remaining to be supressed. Reset type: SYSRSn
15-7	RESERVED	R	0h	Reserved
6-0	NSEL	R/W	0h	ADC Trigger Repeater 2 selection of number of triggers. In oversampling mode, selects the number of repeated triggers. For each trigger received corresponding to REP2CTL.TRIGSEL, NSEL + 1 triggers will be generated. 0 = 1 trigger is generated (pass-through) 1 = 2 triggers are generated 2 = 3 triggers are generated ... 127 = 128 triggers are generated In undersampling mode, selects the number triggers to be supressed. 1 out NSEL + 1 triggers received corresponding to REP2CTL.TRIGSEL will be passed through (the first trigger will be passed through and the subsequent NSEL triggers will be supressed). 0 = all triggers are passed 1 = 1 out of 2 triggers are passed 2 = 1 out of 3 triggers are passed ... 127 = 1 out of 128 triggers are passed Reset type: SYSRSn

24.16.3.108 REP2PHASE Register (Offset = 1BCh) [Reset = 00000000h]

REP2PHASE is shown in Figure 24-211 and described in Table 24-189.

Return to the Summary Table.

ADC Trigger Repeater 2 Phase Select Register

Figure 24-211. REP2PHASE Register

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

Table 24-189. REP2PHASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PHASECOUNT	R	0h	ADC trigger repeater 2 phase delay status. When the trigger selected by REP2CTL.TRIGSEL is received, this register will start counting down from PHASECOUNT until the counter reaches 0, at which point the trigger will be passed on to the repeater re-trigger logic. If the trigger selected by REP2CTL.TRIGSEL is recieved when PHASECOUNT is not 0 (the phase delay logic is busy from the previous trigger) then the new trigger will be ignored and REP2CTL.PHASEOVF will be set to 1. Reset type: SYSRSn
15-0	PHASE	R/W	0h	ADC trigger repeater 2 phase delay configuration. Defines the number of SYSCLKs to delay the selected trigger before passing it on to the re-triggering logic. 0 = trigger is passed through without delay 1 = trigger is delayed by 1 SYSCLK 2 = trigger is delayed by 2 SYSCLKs ... 65535 = trigger is delayed by 65535 SYSCLKs Reset type: SYSRSn

24.16.3.109 REP2SPREAD Register (Offset = 1C0h) [Reset = 00000000h]

REP2SPREAD is shown in Figure 24-212 and described in Table 24-190.

Return to the Summary Table.

ADC Trigger Repeater 2 Spread Select Register

Figure 24-212. REP2SPREAD Register

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

Table 24-190. REP2SPREAD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SPREADCOUNT	R	0h	ADC trigger repeater 2 spread status. When a trigger is sent to the ADC in oversampling mode, this register will start counting down from SPREAD until SPREADCOUNT equals 0. The next repeated trigger to the ADC in oversampling mode will not occur until SPREADCOUNT is 0 (minimum time is complete) and REP2CTL.BUSY = 0 (SOCs associated with trigger repeater 2 are no longer pending). Reset type: SYSRSn
15-0	SPREAD	R/W	0h	ADC trigger repeater 2 spread delay configuration. In oversampling mode, defines the minimum number of SYSCLKs to wait before creating the next repeated trigger to the ADC. If SPREAD is less than the time needed for all SOCs associated with repeater 2 to sample and convert, then the repeater will generate triggers as fast as the ADC can convert the associated conversions. If SPREAD is greater than the time needed for all SOCs associated with repeater 2 to sample and convert, then repeated triggers to the ADC will be SPREAD SYSCLK cycles apart. 0 = oversampled repeated triggers occur as fast as the ADC can sample and convert associated SOCs 1 = time between repeated triggers is at least 1 SYSCLKs 2 = time between repeated triggers is at least 2 SYSCLKs ... 65535 = time between repeated triggers is at least 65535 SYSCLKs Reset type: SYSRSn

24.16.3.110 REP2FRC Register (Offset = 1C4h) [Reset = 0000h]

REP2FRC is shown in Figure 24-213 and described in Table 24-191.

Return to the Summary Table.

ADC Trigger Repeater 2 Software Force Register

Figure 24-213. REP2FRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							SWFRC
R-0h							R-0/W1S-0h

Table 24-191. REP2FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R	0h	Reserved
0	SWFRC	R-0/W1S	0h	Write 1 to force a trigger to repeat block 2 input regardless of the value of TRIGGER. Always reads 0. Reset type: SYSRSn

24.16.3.111 ADCPPB1LIMIT Register (Offset = 1D4h) [Reset = 0000h]

ADCPPB1LIMIT is shown in Figure 24-214 and described in Table 24-192.

Return to the Summary Table.

ADC PPB1Conversion Count Limit Register

Figure 24-214. ADCPPB1LIMIT Register

15	14	13	12	11	10	9	8
RESERVED						LIMIT	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
LIMIT							
R/W-0h							

Table 24-192. ADCPPB1LIMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	LIMIT	R/W	0h	Post Processing Block 1 Oversampling Limit. Defines the number of conversions to accumulate before PSUM is automatically loaded into SUM. To prevent PSUM from overflowing, do not write a value larger than 128 when the ADC is operating in 16-bit mode. Reset type: SYSRSn

24.16.3.112 ADCPPBP1PCOUNT Register (Offset = 1D8h) [Reset = 0000h]

ADCPPBP1PCOUNT is shown in Figure 24-215 and described in Table 24-193.

Return to the Summary Table.

ADC PPB1 Partial Conversion Count Register

Figure 24-215. ADCPPBP1PCOUNT Register

15	14	13	12	11	10	9	8
RESERVED						PCOUNT	
R-0h						R-0h	
7	6	5	4	3	2	1	0
PCOUNT							
R-0h							

Table 24-193. ADCPPBP1PCOUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	PCOUNT	R	0h	Post Processing Block 1 Oversampling Partial Count. Each time a new result propagates through the PPB signal chain and accumulates into ADCPPB1PSUM this register is incremented by 1. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

24.16.3.113 ADCPPB1CONFIG2 Register (Offset = 1DCh) [Reset = 0000h]

ADCPPB1CONFIG2 is shown in Figure 24-216 and described in Table 24-194.

Return to the Summary Table.

ADC PPB1 Sum Shift Register

Figure 24-216. ADCPPB1CONFIG2 Register

15	14	13	12	11	10	9	8
COMPSEL		RESERVED	OSINTSEL	SWSYNC	RESERVED	SYNCINSEL	
R/W-0h		R-0h	R/W-0h	R-0/W1S-0h	R-0h	R/W-0h	
7	6	5	4	3	2	1	0
SYNCINSEL				SHIFT			
R/W-0h				R/W-0h			

Table 24-194. ADCPPB1CONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	COMPSEL	R/W	0h	Post Processing Block 1 Compare Source Select. This field determines whether ADCPPB1RESULT, ADCPPB1PSUM, or ADCPPB1SUM is used for the zero-crossing detect logic and threshold compare. 00 = ADCPPB1RESULT is used for compare logic 01 = ADCPPB1PSUM is used for compare logic 10 = ADCPPB1SUM is used for compare logic 11 = Reserved Note: when ADCPPB1PSUM is selected as the compare source and when a LIMIT match occurs (ADCPPB1LIMIT equals ADCPPB1COUNT) the ADCPPB1PSUM register will be cleared and the final sum will be loaded into ADCPPB1SUM. For this sample, the final sum, ADCPPB1SUM will be used for the compariosn instead of ADCPPB1PSUM. Reset type: SYSRSn
13	RESERVED	R	0h	Reserved
12	OSINTSEL	R/W	0h	Post Processing Block 1 Interrupt Source Select. OSINT1 can be used to trigger an ADC interrupt (ADCINT1 through ADCINT4) via selection in the ADCINT1N2 or ADCINT3N4. This selection determines if a sync. event can trigger OSINT1 in addition to a PCOUNT = LIMIT event. 0 = OSINT1 will be generated from PCOUNT = LIMIT only 1 = OSTIN1 will be generated form PCOUNT = LIMIT or a sync. event. Note: If a SYNC event would cause an OSINT one cycle after OSINT would have been cause by PCOUNT = LIMIT match, then the second OSINT is ignored. Reset type: SYSRSn
11	SWSYNC	R-0/W1S	0h	PPB 1 software force sync. On a sync. event, all partial registers transfer to the final registers and are then reset. Note: In the case where the software force occurs at the same time that a new sample is added to the PSUM and the PSUM is being used for a high or low limit compare, then the comparison will not occur. Reset type: SYSRSn
10	RESERVED	R	0h	Reserved
9-4	SYNCINSEL	R/W	0h	PPB 1 sync. input select. On a sync. event, all partial registers transfer to the final registers and are then reset. Refer to SOC spec for details Reset type: SYSRSn

Table 24-194. ADCPPB1CONFIG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	SHIFT	R/W	0h	Post Processing Block 1 right shift. Defines the number of bits to right shift PSUM before loading into the final SUM. 0 : no right shift 1 : SUM = PSUM >> 1 2 : SUM = PSUM >> 2 ... 10 : SUM = PSUM >> 10 11 - 15 : Reserved Reset type: SYSRSn

24.16.3.114 ADCPPB1PSUM Register (Offset = 1E0h) [Reset = 00000000h]

ADCPPB1PSUM is shown in Figure 24-217 and described in Table 24-195.

Return to the Summary Table.

ADC PPB1 Partial Sum Register

Figure 24-217. ADCPPB1PSUM Register

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

Table 24-195. ADCPPB1PSUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 23. Reset type: SYSRSn
23-0	PSUM	R	0h	Post Processing Block 1 Oversampling Partial Sum. Each time a new result propagates through the PPB signal chain and latches into ADCPPB1RESULT the result is subsequently accumulated into this register. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 2 SYSCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles (refer to the TRM for detailed timing information). Reset type: SYSRSn

24.16.3.115 ADCPPB1PMAX Register (Offset = 1E4h) [Reset = 00000000h]

ADCPPB1PMAX is shown in Figure 24-218 and described in Table 24-196.

Return to the Summary Table.

ADC PPB1 Partial Max Register

Figure 24-218. ADCPPB1PMAX Register

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

Table 24-196. ADCPPB1PMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. Reset type: SYSRSn
16-0	PMAX	R	0h	Post Processing Block 1 Oversampling Partial Max. Each time a new result propagates through the PPB signal chain and latches into ADCPPB1RESULT the result replaces this register if it is larger. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

24.16.3.116 ADCPPB1PMAXI Register (Offset = 1E8h) [Reset = 0000h]

ADCPPB1PMAXI is shown in Figure 24-219 and described in Table 24-197.

Return to the Summary Table.

ADC PPB1 Partial Max Index Register

Figure 24-219. ADCPPB1PMAXI Register

15	14	13	12	11	10	9	8
RESERVED						PMAXI	
R-0h						R-0h	
7	6	5	4	3	2	1	0
PMAXI							
R-0h							

Table 24-197. ADCPPB1PMAXI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	PMAXI	R	0h	Post Processing Block 1 Oversampling Partial Index of Max. Each time a new result propagates through the PPB signal chain and latches into ADCPPB1RESULT if the result replaces PMAX this register is loaded with the current value of PCOUNT. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

24.16.3.117 ADCPPB1PMIN Register (Offset = 1ECh) [Reset = 00000000h]

ADCPPB1PMIN is shown in Figure 24-220 and described in Table 24-198.

Return to the Summary Table.

ADC PPB1 Partial MIN Register

Figure 24-220. ADCPPB1PMIN Register

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

Table 24-198. ADCPPB1PMIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. Reset type: SYSRSn
16-0	PMIN	R	0h	Post Processing Block 1 Oversampling Partial Min. Each time a new result propagates through the PPB signal chain and latches into ADCPPB1RESULT the result replaces this register if it is smaller. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

24.16.3.118 ADCPPB1PMINI Register (Offset = 1F0h) [Reset = 0000h]

ADCPPB1PMINI is shown in Figure 24-221 and described in Table 24-199.

Return to the Summary Table.

ADC PPB1 Partial Min Index Register

Figure 24-221. ADCPPB1PMINI Register

15	14	13	12	11	10	9	8
RESERVED						PMINI	
R-0h						R-0h	
7	6	5	4	3	2	1	0
PMINI							
R-0h							

Table 24-199. ADCPPB1PMINI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	PMINI	R	0h	Post Processing Block 1 Oversampling Partial Index of Min. Each time a new result propagates through the PPB signal chain and latches into ADCPPB1RESULT if the result replaces PMIN this register is loaded with the current value of PCOUNT. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

24.16.3.119 ADCPPB1TRIPLO2 Register (Offset = 1F4h) [Reset = 00000000h]

ADCPPB1TRIPLO2 is shown in Figure 24-222 and described in Table 24-200.

Return to the Summary Table.

ADC PPB1 Extended Trip Low Register

Figure 24-222. ADCPPB1TRIPLO2 Register

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

Table 24-200. ADCPPB1TRIPLO2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	LIMITLO	R/W	0h	ADC Post Processing Block 1 Trip High Low Limit. This value sets the digital comparator trip low limit if ADCPPB1TRIPLO.LIMITLO2EN = 1. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPLO2[23:17] will be ignored in 16 bit mode - TRIPLO2[23:13] will be ignored in 12 bit mode Reset type: SYSRSn

24.16.3.120 ADCPPB2LIMIT Register (Offset = 208h) [Reset = 0000h]

ADCPPB2LIMIT is shown in Figure 24-223 and described in Table 24-201.

Return to the Summary Table.

ADC PPB2Conversion Count Limit Register

Figure 24-223. ADCPPB2LIMIT Register

15	14	13	12	11	10	9	8
RESERVED						LIMIT	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
LIMIT							
R/W-0h							

Table 24-201. ADCPPB2LIMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	LIMIT	R/W	0h	Post Processing Block 2 Oversampling Limit. Defines the number of conversions to accumulate before PSUM is automatically loaded into SUM. To prevent PSUM from overflowing, do not write a value larger than 128 when the ADC is operating in 16-bit mode. Reset type: SYSRSn

24.16.3.121 ADCPPBP2PCOUNT Register (Offset = 20Ch) [Reset = 0000h]

ADCPPBP2PCOUNT is shown in Figure 24-224 and described in Table 24-202.

Return to the Summary Table.

ADC PPB2 Partial Conversion Count Register

Figure 24-224. ADCPPBP2PCOUNT Register

15	14	13	12	11	10	9	8
RESERVED						PCOUNT	
R-0h						R-0h	
7	6	5	4	3	2	1	0
PCOUNT							
R-0h							

Table 24-202. ADCPPBP2PCOUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	PCOUNT	R	0h	Post Processing Block 2 Oversampling Partial Count. Each time a new result propagates through the PPB signal chain and accumulates into ADCPPB2PSUM this register is incremented by 1. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB2RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

24.16.3.122 ADCPPB2CONFIG2 Register (Offset = 210h) [Reset = 0000h]

ADCPPB2CONFIG2 is shown in Figure 24-225 and described in Table 24-203.

Return to the Summary Table.

ADC PPB2 Sum Shift Register

Figure 24-225. ADCPPB2CONFIG2 Register

15	14	13	12	11	10	9	8
COMPSEL		RESERVED	OSINTSEL	SWSYNC	RESERVED	SYNCINSEL	
R/W-0h		R-0h	R/W-0h	R-0/W1S-0h	R-0h	R/W-0h	
7	6	5	4	3	2	1	0
SYNCINSEL				SHIFT			
R/W-0h				R/W-0h			

Table 24-203. ADCPPB2CONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	COMPSEL	R/W	0h	Post Processing Block 2 Compare Source Select. This field determines whether ADCPPB2RESULT, ADCPPB2PSUM, or ADCPPB2SUM is used for the zero-crossing detect logic and threshold compare. 00 = ADCPPB2RESULT is used for compare logic 01 = ADCPPB2PSUM is used for compare logic 10 = ADCPPB2SUM is used for compare logic 11 = Reserved Note: when ADCPPB2PSUM is selected as the compare source and when a LIMIT match occurs (ADCPPB2LIMIT equals ADCPPB2COUNT) the ADCPPB2PSUM register will be cleared and the final sum will be loaded into ADCPPB2SUM. For this sample, the final sum, ADCPPB2SUM will be used for the compariosn instead of ADCPPB2PSUM. Reset type: SYSRSn
13	RESERVED	R	0h	Reserved
12	OSINTSEL	R/W	0h	Post Processing Block 2 Interrupt Source Select. OSINT2 can be used to trigger an ADC interrupt (ADCINT1 through ADCINT4) via selection in the ADCINT1N2 or ADCINT3N4. This selection determines if a sync. event can trigger OSINT2 in addition to a PCOUNT = LIMIT event. 0 = OSINT2 will be generated from PCOUNT = LIMIT only 1 = OSTIN2 will be generated form PCOUNT = LIMIT or a sync. event. Note: If a SYNC event would cause an OSINT one cycle after OSINT would have been cause by PCOUNT = LIMIT match, then the second OSINT is ignored. Reset type: SYSRSn
11	SWSYNC	R-0/W1S	0h	PPB 2 software force sync. On a sync. event, all partial registers transfer to the final registers and are then reset. Note: In the case where the software force occurs at the same time that a new sample is added to the PSUM and the PSUM is being used for a high or low limit compare, then the comparison will not occur. Reset type: SYSRSn
10	RESERVED	R	0h	Reserved
9-4	SYNCINSEL	R/W	0h	PPB 2 sync. input select. On a sync. event, all partial registers transfer to the final registers and are then reset. Refer to SOC spec for details Reset type: SYSRSn

Table 24-203. ADCPPB2CONFIG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	SHIFT	R/W	0h	Post Processing Block 2 right shift. Defines the number of bits to right shift PSUM before loading into the final SUM. 0 : no right shift 1 : SUM = PSUM >> 1 2 : SUM = PSUM >> 2 ... 10 : SUM = PSUM >> 10 11 - 15 : Reserved Reset type: SYSRSn

24.16.3.123 ADCPPB2PSUM Register (Offset = 214h) [Reset = 00000000h]

ADCPPB2PSUM is shown in Figure 24-226 and described in Table 24-204.

Return to the Summary Table.

ADC PPB2 Partial Sum Register

Figure 24-226. ADCPPB2PSUM Register

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

Table 24-204. ADCPPB2PSUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 23. Reset type: SYSRSn
23-0	PSUM	R	0h	Post Processing Block 2 Oversampling Partial Sum. Each time a new result propagates through the PPB signal chain and latches into ADCPPB2RESULT the result is subsequently accumulated into this register. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB2RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 2 SYSCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles (refer to the TRM for detailed timing information). Reset type: SYSRSn

24.16.3.124 ADCPPB2PMAX Register (Offset = 218h) [Reset = 00000000h]

ADCPPB2PMAX is shown in Figure 24-227 and described in Table 24-205.

Return to the Summary Table.

ADC PPB2 Partial Max Register

Figure 24-227. ADCPPB2PMAX Register

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

Table 24-205. ADCPPB2PMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. Reset type: SYSRSn
16-0	PMAX	R	0h	Post Processing Block 2 Oversampling Partial Max. Each time a new result propagates through the PPB signal chain and latches into ADCPPB2RESULT the result replaces this register if it is larger. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB2RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

24.16.3.125 ADCPPB2PMAXI Register (Offset = 21Ch) [Reset = 0000h]

ADCPPB2PMAXI is shown in Figure 24-228 and described in Table 24-206.

Return to the Summary Table.

ADC PPB2 Partial Max Index Register

Figure 24-228. ADCPPB2PMAXI Register

15	14	13	12	11	10	9	8
RESERVED						PMAXI	
R-0h						R-0h	
7	6	5	4	3	2	1	0
PMAXI							
R-0h							

Table 24-206. ADCPPB2PMAXI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	PMAXI	R	0h	Post Processing Block 2 Oversampling Partial Index of Max. Each time a new result propagates through the PPB signal chain and latches into ADCPPB2RESULT if the result replaces PMAX this register is loaded with the current value of PCOUNT. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB2RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

24.16.3.126 ADCPPB2PMIN Register (Offset = 220h) [Reset = 00000000h]

ADCPPB2PMIN is shown in Figure 24-229 and described in Table 24-207.

Return to the Summary Table.

ADC PPB2 Partial MIN Register

Figure 24-229. ADCPPB2PMIN Register

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

Table 24-207. ADCPPB2PMIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. Reset type: SYSRSn
16-0	PMIN	R	0h	Post Processing Block 2 Oversampling Partial Min. Each time a new result propagates through the PPB signal chain and latches into ADCPPB2RESULT the result replaces this register if it is smaller. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB2RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

24.16.3.127 ADCPPB2PMINI Register (Offset = 224h) [Reset = 0000h]

ADCPPB2PMINI is shown in Figure 24-230 and described in Table 24-208.

Return to the Summary Table.

ADC PPB2 Partial Min Index Register

Figure 24-230. ADCPPB2PMINI Register

15	14	13	12	11	10	9	8
RESERVED						PMINI	
R-0h						R-0h	
7	6	5	4	3	2	1	0
PMINI							
R-0h							

Table 24-208. ADCPPB2PMINI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	PMINI	R	0h	Post Processing Block 2 Oversampling Partial Index of Min. Each time a new result propagates through the PPB signal chain and latches into ADCPPB2RESULT if the result replaces PMIN this register is loaded with the current value of PCOUNT. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB2RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

24.16.3.128 ADCPPB2TRIPLO2 Register (Offset = 228h) [Reset = 00000000h]

ADCPPB2TRIPLO2 is shown in Figure 24-231 and described in Table 24-209.

Return to the Summary Table.

ADC PPB2 Extended Trip Low Register

Figure 24-231. ADCPPB2TRIPLO2 Register

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

Table 24-209. ADCPPB2TRIPLO2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	LIMITLO	R/W	0h	ADC Post Processing Block 2 Trip High Low Limit. This value sets the digital comparator trip low limit if ADCPPB2TRIPLO.LIMITLO2EN = 1. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPLO2[23:17] will be ignored in 16 bit mode - TRIPLO2[23:13] will be ignored in 12 bit mode Reset type: SYSRSn

24.16.3.129 ADCPPB3LIMIT Register (Offset = 23Ch) [Reset = 0000h]

ADCPPB3LIMIT is shown in Figure 24-232 and described in Table 24-210.

Return to the Summary Table.

ADC PPB3Conversion Count Limit Register

Figure 24-232. ADCPPB3LIMIT Register

15	14	13	12	11	10	9	8
RESERVED						LIMIT	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
LIMIT							
R/W-0h							

Table 24-210. ADCPPB3LIMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	LIMIT	R/W	0h	Post Processing Block 3 Oversampling Limit. Defines the number of conversions to accumulate before PSUM is automatically loaded into SUM. To prevent PSUM from overflowing, do not write a value larger than 128 when the ADC is operating in 16-bit mode. Reset type: SYSRSn

24.16.3.130 ADCPPBP3PCOUNT Register (Offset = 240h) [Reset = 0000h]

ADCPPBP3PCOUNT is shown in Figure 24-233 and described in Table 24-211.

Return to the Summary Table.

ADC PPB3 Partial Conversion Count Register

Figure 24-233. ADCPPBP3PCOUNT Register

15	14	13	12	11	10	9	8
RESERVED						PCOUNT	
R-0h						R-0h	
7	6	5	4	3	2	1	0
PCOUNT							
R-0h							

Table 24-211. ADCPPBP3PCOUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	PCOUNT	R	0h	Post Processing Block 3 Oversampling Partial Count. Each time a new result propagates through the PPB signal chain and accumulates into ADCPPB3PSUM this register is incremented by 1. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB3RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

24.16.3.131 ADCPPB3CONFIG2 Register (Offset = 244h) [Reset = 0000h]

ADCPPB3CONFIG2 is shown in Figure 24-234 and described in Table 24-212.

Return to the Summary Table.

ADC PPB3 Sum Shift Register

Figure 24-234. ADCPPB3CONFIG2 Register

15	14	13	12	11	10	9	8
COMPSEL		RESERVED	OSINTSEL	SWSYNC	RESERVED	SYNCINSEL	
R/W-0h		R-0h	R/W-0h	R-0/W1S-0h	R-0h	R/W-0h	
7	6	5	4	3	2	1	0
SYNCINSEL				SHIFT			
R/W-0h				R/W-0h			

Table 24-212. ADCPPB3CONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	COMPSEL	R/W	0h	Post Processing Block 3 Compare Source Select. This field determines whether ADCPPB3RESULT, ADCPPB3PSUM, or ADCPPB3SUM is used for the zero-crossing detect logic and threshold compare. 00 = ADCPPB3RESULT is used for compare logic 01 = ADCPPB3PSUM is used for compare logic 10 = ADCPPB3SUM is used for compare logic 11 = Reserved Note: when ADCPPB3PSUM is selected as the compare source and when a LIMIT match occurs (ADCPPB3LIMIT equals ADCPPB3COUNT) the ADCPPB3PSUM register will be cleared and the final sum will be loaded into ADCPPB3SUM. For this sample, the final sum, ADCPPB3SUM will be used for the compariosn instead of ADCPPB3PSUM. Reset type: SYSRSn
13	RESERVED	R	0h	Reserved
12	OSINTSEL	R/W	0h	Post Processing Block 3 Interrupt Source Select. OSINT3 can be used to trigger an ADC interrupt (ADCINT1 through ADCINT4) via selection in the ADCINT1N2 or ADCINT3N4. This selection determines if a sync. event can trigger OSINT3 in addition to a PCOUNT = LIMIT event. 0 = OSINT3 will be generated from PCOUNT = LIMIT only 1 = OSTIN3 will be generated form PCOUNT = LIMIT or a sync. event. Note: If a SYNC event would cause an OSINT one cycle after OSINT would have been cause by PCOUNT = LIMIT match, then the second OSINT is ignored. Reset type: SYSRSn
11	SWSYNC	R-0/W1S	0h	PPB 3 software force sync. On a sync. event, all partial registers transfer to the final registers and are then reset. Note: In the case where the software force occurs at the same time that a new sample is added to the PSUM and the PSUM is being used for a high or low limit compare, then the comparison will not occur. Reset type: SYSRSn
10	RESERVED	R	0h	Reserved
9-4	SYNCINSEL	R/W	0h	PPB 3 sync. input select. On a sync. event, all partial registers transfer to the final registers and are then reset. Refer to SOC spec for details Reset type: SYSRSn

Table 24-212. ADCPPB3CONFIG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	SHIFT	R/W	0h	Post Processing Block 3 right shift. Defines the number of bits to right shift PSUM before loading into the final SUM. 0 : no right shift 1 : SUM = PSUM >> 1 2 : SUM = PSUM >> 2 ... 10 : SUM = PSUM >> 10 11 - 15 : Reserved Reset type: SYSRSn

24.16.3.132 ADCPPB3PSUM Register (Offset = 248h) [Reset = 00000000h]

ADCPPB3PSUM is shown in Figure 24-235 and described in Table 24-213.

Return to the Summary Table.

ADC PPB3 Partial Sum Register

Figure 24-235. ADCPPB3PSUM Register

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

Table 24-213. ADCPPB3PSUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 23. Reset type: SYSRSn
23-0	PSUM	R	0h	Post Processing Block 3 Oversampling Partial Sum. Each time a new result propagates through the PPB signal chain and latches into ADCPPB3RESULT the result is subsequently accumulated into this register. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB3RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 2 SYSCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles (refer to the TRM for detailed timing information). Reset type: SYSRSn

24.16.3.133 ADCPPB3PMAX Register (Offset = 24Ch) [Reset = 00000000h]

ADCPPB3PMAX is shown in Figure 24-236 and described in Table 24-214.

Return to the Summary Table.

ADC PPB3 Partial Max Register

Figure 24-236. ADCPPB3PMAX Register

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

Table 24-214. ADCPPB3PMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. Reset type: SYSRSn
16-0	PMAX	R	0h	Post Processing Block 3 Oversampling Partial Max. Each time a new result propagates through the PPB signal chain and latches into ADCPPB3RESULT the result replaces this register if it is larger. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB3RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

24.16.3.134 ADCPPB3PMAXI Register (Offset = 250h) [Reset = 0000h]

ADCPPB3PMAXI is shown in Figure 24-237 and described in Table 24-215.

Return to the Summary Table.

ADC PPB3 Partial Max Index Register

Figure 24-237. ADCPPB3PMAXI Register

15	14	13	12	11	10	9	8
RESERVED						PMAXI	
R-0h						R-0h	
7	6	5	4	3	2	1	0
PMAXI							
R-0h							

Table 24-215. ADCPPB3PMAXI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	PMAXI	R	0h	Post Processing Block 3 Oversampling Partial Index of Max. Each time a new result propagates through the PPB signal chain and latches into ADCPPB3RESULT if the result replaces PMAX this register is loaded with the current value of PCOUNT. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB3RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

24.16.3.135 ADCPPB3PMIN Register (Offset = 254h) [Reset = 00000000h]

ADCPPB3PMIN is shown in Figure 24-238 and described in Table 24-216.

Return to the Summary Table.

ADC PPB3 Partial MIN Register

Figure 24-238. ADCPPB3PMIN Register

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

Table 24-216. ADCPPB3PMIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. Reset type: SYSRSn
16-0	PMIN	R	0h	Post Processing Block 3 Oversampling Partial Min. Each time a new result propagates through the PPB signal chain and latches into ADCPPB3RESULT the result replaces this register if it is smaller. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB3RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

24.16.3.136 ADCPPB3PMINI Register (Offset = 258h) [Reset = 0000h]

ADCPPB3PMINI is shown in Figure 24-239 and described in Table 24-217.

Return to the Summary Table.

ADC PPB3 Partial Min Index Register

Figure 24-239. ADCPPB3PMINI Register

15	14	13	12	11	10	9	8
RESERVED						PMINI	
R-0h						R-0h	
7	6	5	4	3	2	1	0
PMINI							
R-0h							

Table 24-217. ADCPPB3PMINI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	PMINI	R	0h	Post Processing Block 3 Oversampling Partial Index of Min. Each time a new result propagates through the PPB signal chain and latches into ADCPPB3RESULT if the result replaces PMIN this register is loaded with the current value of PCOUNT. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB3RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

24.16.3.137 ADCPPB3TRIPLO2 Register (Offset = 25Ch) [Reset = 00000000h]

ADCPPB3TRIPLO2 is shown in Figure 24-240 and described in Table 24-218.

Return to the Summary Table.

ADC PPB3 Extended Trip Low Register

Figure 24-240. ADCPPB3TRIPLO2 Register

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

Table 24-218. ADCPPB3TRIPLO2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	LIMITLO	R/W	0h	ADC Post Processing Block 3 Trip High Low Limit. This value sets the digital comparator trip low limit if ADCPPB3TRIPLO.LIMITLO2EN = 1. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPLO2[23:17] will be ignored in 16 bit mode - TRIPLO2[23:13] will be ignored in 12 bit mode Reset type: SYSRSn

24.16.3.138 ADCPPB4LIMIT Register (Offset = 270h) [Reset = 0000h]

ADCPPB4LIMIT is shown in Figure 24-241 and described in Table 24-219.

Return to the Summary Table.

ADC PPB4Conversion Count Limit Register

Figure 24-241. ADCPPB4LIMIT Register

15	14	13	12	11	10	9	8
RESERVED						LIMIT	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
LIMIT							
R/W-0h							

Table 24-219. ADCPPB4LIMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	LIMIT	R/W	0h	Post Processing Block 4 Oversampling Limit. Defines the number of conversions to accumulate before PSUM is automatically loaded into SUM. To prevent PSUM from overflowing, do not write a value larger than 128 when the ADC is operating in 16-bit mode. Reset type: SYSRSn

24.16.3.139 ADCPPBP4PCOUNT Register (Offset = 274h) [Reset = 0000h]

ADCPPBP4PCOUNT is shown in Figure 24-242 and described in Table 24-220.

Return to the Summary Table.

ADC PPB4 Partial Conversion Count Register

Figure 24-242. ADCPPBP4PCOUNT Register

15	14	13	12	11	10	9	8
RESERVED						PCOUNT	
R-0h						R-0h	
7	6	5	4	3	2	1	0
PCOUNT							
R-0h							

Table 24-220. ADCPPBP4PCOUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	PCOUNT	R	0h	Post Processing Block 4 Oversampling Partial Count. Each time a new result propagates through the PPB signal chain and accumulates into ADCPPB4PSUM this register is incremented by 1. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB4RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

24.16.3.140 ADCPPB4CONFIG2 Register (Offset = 278h) [Reset = 0000h]

ADCPPB4CONFIG2 is shown in Figure 24-243 and described in Table 24-221.

Return to the Summary Table.

ADC PPB4 Sum Shift Register

Figure 24-243. ADCPPB4CONFIG2 Register

15	14	13	12	11	10	9	8
COMPSEL		RESERVED	OSINTSEL	SWSYNC	RESERVED	SYNCINSEL	
R/W-0h		R-0h	R/W-0h	R-0/W1S-0h	R-0h	R/W-0h	
7	6	5	4	3	2	1	0
SYNCINSEL				SHIFT			
R/W-0h				R/W-0h			

Table 24-221. ADCPPB4CONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	COMPSEL	R/W	0h	Post Processing Block 4 Compare Source Select. This field determines whether ADCPPB4RESULT, ADCPPB4PSUM, or ADCPPB4SUM is used for the zero-crossing detect logic and threshold compare. 00 = ADCPPB4RESULT is used for compare logic 01 = ADCPPB4PSUM is used for compare logic 10 = ADCPPB4SUM is used for compare logic 11 = Reserved Note: when ADCPPB4PSUM is selected as the compare source and when a LIMIT match occurs (ADCPPB4LIMIT equals ADCPPB4COUNT) the ADCPPB4PSUM register will be cleared and the final sum will be loaded into ADCPPB4SUM. For this sample, the final sum, ADCPPB4SUM will be used for the compariosn instead of ADCPPB4PSUM. Reset type: SYSRSn
13	RESERVED	R	0h	Reserved
12	OSINTSEL	R/W	0h	Post Processing Block 4 Interrupt Source Select. OSINT4 can be used to trigger an ADC interrupt (ADCINT1 through ADCINT4) via selection in the ADCINT1N2 or ADCINT3N4. This selection determines if a sync. event can trigger OSINT4 in addition to a PCOUNT = LIMIT event. 0 = OSINT4 will be generated from PCOUNT = LIMIT only 1 = OSTIN4 will be generated form PCOUNT = LIMIT or a sync. event. Note: If a SYNC event would cause an OSINT one cycle after OSINT would have been cause by PCOUNT = LIMIT match, then the second OSINT is ignored. Reset type: SYSRSn
11	SWSYNC	R-0/W1S	0h	PPB 4 software force sync. On a sync. event, all partial registers transfer to the final registers and are then reset. Note: In the case where the software force occurs at the same time that a new sample is added to the PSUM and the PSUM is being used for a high or low limit compare, then the comparison will not occur. Reset type: SYSRSn
10	RESERVED	R	0h	Reserved
9-4	SYNCINSEL	R/W	0h	PPB 4 sync. input select. On a sync. event, all partial registers transfer to the final registers and are then reset. Refer to SOC spec for details Reset type: SYSRSn

Table 24-221. ADCPPB4CONFIG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	SHIFT	R/W	0h	Post Processing Block 4 right shift. Defines the number of bits to right shift PSUM before loading into the final SUM. 0 : no right shift 1 : SUM = PSUM >> 1 2 : SUM = PSUM >> 2 ... 10 : SUM = PSUM >> 10 11 - 15 : Reserved Reset type: SYSRSn

24.16.3.141 ADCPPB4PSUM Register (Offset = 27Ch) [Reset = 00000000h]

ADCPPB4PSUM is shown in Figure 24-244 and described in Table 24-222.

Return to the Summary Table.

ADC PPB4 Partial Sum Register

Figure 24-244. ADCPPB4PSUM Register

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

Table 24-222. ADCPPB4PSUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 23. Reset type: SYSRSn
23-0	PSUM	R	0h	Post Processing Block 4 Oversampling Partial Sum. Each time a new result propagates through the PPB signal chain and latches into ADCPPB4RESULT the result is subsequently accumulated into this register. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB4RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 2 SYSCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles (refer to the TRM for detailed timing information). Reset type: SYSRSn

24.16.3.142 ADCPPB4PMAX Register (Offset = 280h) [Reset = 00000000h]

ADCPPB4PMAX is shown in Figure 24-245 and described in Table 24-223.

Return to the Summary Table.

ADC PPB4 Partial Max Register

Figure 24-245. ADCPPB4PMAX Register

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

Table 24-223. ADCPPB4PMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. Reset type: SYSRSn
16-0	PMAX	R	0h	Post Processing Block 4 Oversampling Partial Max. Each time a new result propagates through the PPB signal chain and latches into ADCPPB4RESULT the result replaces this register if it is larger. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB4RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

24.16.3.143 ADCPPB4PMAXI Register (Offset = 284h) [Reset = 0000h]

ADCPPB4PMAXI is shown in Figure 24-246 and described in Table 24-224.

Return to the Summary Table.

ADC PPB4 Partial Max Index Register

Figure 24-246. ADCPPB4PMAXI Register

15	14	13	12	11	10	9	8
RESERVED						PMAXI	
R-0h						R-0h	
7	6	5	4	3	2	1	0
PMAXI							
R-0h							

Table 24-224. ADCPPB4PMAXI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	PMAXI	R	0h	Post Processing Block 4 Oversampling Partial Index of Max. Each time a new result propagates through the PPB signal chain and latches into ADCPPB4RESULT if the result replaces PMAX this register is loaded with the current value of PCOUNT. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB4RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

24.16.3.144 ADCPPB4PMIN Register (Offset = 288h) [Reset = 00000000h]

ADCPPB4PMIN is shown in Figure 24-247 and described in Table 24-225.

Return to the Summary Table.

ADC PPB4 Partial MIN Register

Figure 24-247. ADCPPB4PMIN Register

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

Table 24-225. ADCPPB4PMIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 16. Reset type: SYSRSn
16-0	PMIN	R	0h	Post Processing Block 4 Oversampling Partial Min. Each time a new result propagates through the PPB signal chain and latches into ADCPPB4RESULT the result replaces this register if it is smaller. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB4RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

24.16.3.145 ADCPPB4PMINI Register (Offset = 28Ch) [Reset = 0000h]

ADCPPB4PMINI is shown in Figure 24-248 and described in Table 24-226.

Return to the Summary Table.

ADC PPB4 Partial Min Index Register

Figure 24-248. ADCPPB4PMINI Register

15	14	13	12	11	10	9	8
RESERVED						PMINI	
R-0h						R-0h	
7	6	5	4	3	2	1	0
PMINI							
R-0h							

Table 24-226. ADCPPB4PMINI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	PMINI	R	0h	Post Processing Block 4 Oversampling Partial Index of Min. Each time a new result propagates through the PPB signal chain and latches into ADCPPB4RESULT if the result replaces PMIN this register is loaded with the current value of PCOUNT. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB4RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

24.16.3.146 ADCPPB4TRIPLO2 Register (Offset = 290h) [Reset = 00000000h]

ADCPPB4TRIPLO2 is shown in Figure 24-249 and described in Table 24-227.

Return to the Summary Table.

ADC PPB4 Extended Trip Low Register

Figure 24-249. ADCPPB4TRIPLO2 Register

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

Table 24-227. ADCPPB4TRIPLO2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	LIMITLO	R/W	0h	ADC Post Processing Block 4 Trip High Low Limit. This value sets the digital comparator trip low limit if ADCPPB4TRIPLO.LIMITLO2EN = 1. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPLO2[23:17] will be ignored in 16 bit mode - TRIPLO2[23:13] will be ignored in 12 bit mode Reset type: SYSRSn

24.16.4 ADC_SAFECHECK_REGS Registers

Table 24-228 lists the memory-mapped registers for the ADC_SAFECHECK_REGS registers. All register offset addresses not listed in Table 24-228 should be considered as reserved locations and the register contents should not be modified.

Table 24-228. ADC_SAFECHECK_REGS Registers

Offset	Acronym	Register Name	Protection
0h	CHECKCONFIG	ADC Check Configuration Register	
4h	CHECKSTATUS	ADC Check Status Register	
8h	ADCRESSEL1	ADC Check 1 Select Register	
Ch	ADCRESSEL2	ADC Check 2 Select Register	
10h	TOLERANCE	ADC Check Tolerance Register	
18h	CHECKRESULT1	ADC Check Captured Result 1	
1Ch	CHECKRESULT2	ADC Check Captured Result 2	

Complex bit access types are encoded to fit into small table cells. Table 24-229 shows the codes that are used for access types in this section.

Table 24-229. ADC_SAFECHECK_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

24.16.4.1 CHECKCONFIG Register (Offset = 0h) [Reset = 0000h]

CHECKCONFIG is shown in Figure 24-250 and described in Table 24-230.

Return to the Summary Table.

ADC Check Configuration Register

Figure 24-250. CHECKCONFIG Register

15	14	13	12	11	10	9	8
CHKEN	RESERVED						
R/W-0h	R-0h						
7	6	5	4	3	2	1	0
RESERVED	SWSYNC	RESERVED	RESERVED				
R-0h	R-0/W1S-0h	R-0h	R/W-0h				

Table 24-230. CHECKCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
15	CHKEN	R/W	0h	Result Safe Check Module enable Reset type: SYSRSn
14-7	RESERVED	R	0h	Reserved
6	SWSYNC	R-0/W1S	0h	Result Safe Check SW Force Sync. Reset type: SYSRSn
5	RESERVED	R	0h	Reserved
4-0	RESERVED	R/W	0h	Reserved

24.16.4.2 CHECKSTATUS Register (Offset = 4h) [Reset = 0000h]

CHECKSTATUS is shown in Figure 24-251 and described in Table 24-231.

Return to the Summary Table.

ADC Check Status Register

Figure 24-251. CHECKSTATUS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					OOT	RES2READY	RES1READY
R-0h					R-0h	R-0h	R-0h

Table 24-231. CHECKSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-3	RESERVED	R	0h	Reserved
2	OOT	R	0h	Set when the difference between CHECKRESULT1 and CHECKRESULT2 is greater than TOLERANCE after both results have arrived. When set, further results will not be captured into the safety checker module CHECKRESULT1 or CHECKRESULT2 registers and further OOT or OVF events can't be generated. Cleared when the associated OOTx flag for all CPUs are either cleared (via the OOTFLGCLR.OOTx bit) or disabled to all ISR and events (via the CHECKINTSEL3.OOTx, CHECKEVT1SEL3.OOTx, CHECKEVT2SEL3.OOTx, CHECKEVT3SEL3.OOTx, and CHECKEVT4SEL3.OOTx registers) Reset type: SYSRSn
1	RES2READY	R	0h	Result Safe Check Result 2 arrived. Cleared automatically when both results have arrived and the comparison occurs. Can also be cleared by issuing a software sync to the tile via the CHECKCONFIG.SWSYNC field. Reset type: SYSRSn
0	RES1READY	R	0h	Result Safe Check Result 1 arrived. Cleared automatically when both results have arrived and the comparison occurs. Can also be cleared by issuing a software sync to the tile via the CHECKCONFIG.SWSYNC field. Reset type: SYSRSn

24.16.4.3 ADCRESSEL1 Register (Offset = 8h) [Reset = 0000h]

ADCRESSEL1 is shown in Figure 24-252 and described in Table 24-232.

Return to the Summary Table.

ADC Check 1 Select Register

Figure 24-252. ADCRESSEL1 Register

15	14	13	12	11	10	9	8
RESERVED						ADCRESULTSEL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
ADCRESULTSEL				RESERVED		ADCSEL	
R/W-0h				R-0h		R/W-0h	

Table 24-232. ADCRESSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-4	ADCRESULTSEL	R/W	0h	ADC Result Safety Checker Result Select 1 0 = ADCRESULT0 1 = ADCRESULT1 2= ADCRESULT2 3 = ADCRESULT3 4 = ADCRESULT4 5 = ADCRESULT5 6 = ADCRESULT6 7 = ADCRESULT7 8 = ADCRESULT8 9 = ADCRESULT9 10 = ADCRESULT10 11 = ADCRESULT11 12 = ADCRESULT12 13 = ADCRESULT13 14 = ADCRESULT14 15 = ADCRESULT15 16 = ADCRESULT16 17 = ADCRESULT17 18 = ADCRESULT18 19 = ADCRESULT19 20 = ADCRESULT20 21 = ADCRESULT21 22 = ADCRESULT22 23 = ADCRESULT23 24 = ADCRESULT24 25 = ADCRESULT25 26 = ADCRESULT26 27 = ADCRESULT27 28 = ADCRESULT28 29 = ADCRESULT29 30 = ADCRESULT30 31 = ADCRESULT31 32 = ADCPPBRESULT1 33 = ADCPPBRESULT2 34 = ADCPPBRESULT3 35 = ADCPPBRESULT4 36 = ADCPPBSUM1 37 = ADCPPBSUM2 38 = ADCPPBSUM3 39 = ADCPPBSUM4 ... 40 - 61 = Reserved Reset type: SYSRSn

Table 24-232. ADCRESSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	RESERVED	R	0h	Reserved
2-0	ADCSEL	R/W	0h	ADC Result Safety Checker ADC Select 1 0 = ADC-A 1 = ADC-B 2 = ADC-C 3 = ADC-D 4 - 7 = Reserved Reset type: SYSRSn

24.16.4.4 ADCRESSEL2 Register (Offset = Ch) [Reset = 0000h]

ADCRESSEL2 is shown in Figure 24-253 and described in Table 24-233.

Return to the Summary Table.

ADC Check 2 Select Register

Figure 24-253. ADCRESSEL2 Register

15	14	13	12	11	10	9	8
RESERVED						ADCRESULTSEL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
ADCRESULTSEL				RESERVED		ADCSEL	
R/W-0h				R-0h		R/W-0h	

Table 24-233. ADCRESSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-4	ADCRESULTSEL	R/W	0h	ADC Result Safety Checker Result Select 2 0 = ADCRESULT0 1 = ADCRESULT1 2= ADCRESULT2 3 = ADCRESULT3 4 = ADCRESULT4 5 = ADCRESULT5 6 = ADCRESULT6 7 = ADCRESULT7 8 = ADCRESULT8 9 = ADCRESULT9 10 = ADCRESULT10 11 = ADCRESULT11 12 = ADCRESULT12 13 = ADCRESULT13 14 = ADCRESULT14 15 = ADCRESULT15 16 = ADCRESULT16 17 = ADCRESULT17 18 = ADCRESULT18 19 = ADCRESULT19 20 = ADCRESULT20 21 = ADCRESULT21 22 = ADCRESULT22 23 = ADCRESULT23 24 = ADCRESULT24 25 = ADCRESULT25 26 = ADCRESULT26 27 = ADCRESULT27 28 = ADCRESULT28 29 = ADCRESULT29 30 = ADCRESULT30 31 = ADCRESULT31 32 = ADCPPBRESULT1 33 = ADCPPBRESULT2 34 = ADCPPBRESULT3 35 = ADCPPBRESULT4 36 = ADCPPBSUM1 37 = ADCPPBSUM2 38 = ADCPPBSUM3 39 = ADCPPBSUM4 ... 40 - 61 = Reserved Reset type: SYSRSn

Table 24-233. ADCRESSEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	RESERVED	R	0h	Reserved
2-0	ADCSEL	R/W	0h	ADC Result Safety Checker ADC Select 2 0 = ADC-A 1 = ADC-B 2 = ADC-C 3 = ADC-D 4 - 7 = Reserved Reset type: SYSRSn

24.16.4.5 TOLERANCE Register (Offset = 10h) [Reset = 00000000h]

TOLERANCE is shown in Figure 24-254 and described in Table 24-234.

Return to the Summary Table.

ADC Check Tolerance Register

Figure 24-254. TOLERANCE Register

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

Table 24-234. TOLERANCE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	TOLERANCE	R/W	0h	Tolerance for the difference between CHECKRESULT1 and CHECKRESULT2. If the difference is greater than (but not equal to) the tolerance, an out-of-tolerance event will be generated, indicated the compared ADC results are not within expected tolerance of each other. Reset type: SYSRSn

24.16.4.6 CHECKRESULT1 Register (Offset = 18h) [Reset = 00000000h]

CHECKRESULT1 is shown in Figure 24-255 and described in Table 24-235.

Return to the Summary Table.

ADC Check Captured Result 1

Figure 24-255. CHECKRESULT1 Register

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

Table 24-235. CHECKRESULT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	RESULT	R	0h	ADC Result Safety Checker Captured Result Result that was captured In the case that multiple results arrive for one selected result before one result arrives for the other result for comparison, the RES1OVF flag in CHECKSTATUS will be set. This does not prevent CHECKRESULT1 from updating to the latest result. Reset type: SYSRSn

24.16.4.7 CHECKRESULT2 Register (Offset = 1Ch) [Reset = 00000000h]

CHECKRESULT2 is shown in Figure 24-256 and described in Table 24-236.

Return to the Summary Table.

ADC Check Captured Result 2

Figure 24-256. CHECKRESULT2 Register

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

Table 24-236. CHECKRESULT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	RESULT	R	0h	ADC Result Safety Checker Captured Result Result that was captured In the case that multiple results arrive for one selected result before one result arrives for the other result for comparison, the RES2OVF flag in CHECKSTATUS will be set. This does not prevent CHECKRESULT2 from updating to the latest result. Reset type: SYSRSn

24.16.5 ADC_SAFECHECK_INTEVT_REGS Registers

Table 24-237 lists the memory-mapped registers for the ADC_SAFECHECK_INTEVT_REGS registers. All register offset addresses not listed in Table 24-237 should be considered as reserved locations and the register contents should not be modified.

Table 24-237. ADC_SAFECHECK_INTEVT_REGS Registers

Offset	Acronym	Register Name	Protection
0h	OOTFLG	Checker Out-of-Tolerance Flag Register	
4h	OOTFLGCLR	Checker Out-of-Tolerance Flag Clear Register	
8h	RES1OVF	Checker Overflow Result 1 Flag Register	
Ch	RES1OVFCLR	Checker Overflow Result 1 Flag Clear Register	
10h	RES2OVF	Checker Overflow Result 2 Flag Register	
14h	RES2OVFCLR	Checker Overflow Result 2 Flag Clear Register	
18h	CHECKINTFLG	Checker Interrupt Flag Register	
1Ch	CHECKINTFLGCLR	Checker Interrupt Flag Clear Register	
20h	CHECKINTSEL1	Checker Interrupt Source Select Register 1	
24h	CHECKINTSEL2	Checker Interrupt Source Select Register 2	
28h	CHECKINTSEL3	Checker Interrupt Source Select Register 3	
30h	CHECKEVT1SEL1	Checker X-Bar EVT1 Source Select Register 1	
34h	CHECKEVT1SEL2	Checker X-Bar EVT1 Source Select Register 2	
38h	CHECKEVT1SEL3	Checker X-Bar EVT1 Source Select Register 3	
40h	CHECKEVT2SEL1	Checker X-Bar EVT2 Source Select Register 1	
44h	CHECKEVT2SEL2	Checker X-Bar EVT2 Source Select Register 2	
48h	CHECKEVT2SEL3	Checker X-Bar EVT2 Source Select Register 3	
50h	CHECKEVT3SEL1	Checker X-Bar EVT3 Source Select Register 1	
54h	CHECKEVT3SEL2	Checker X-Bar EVT3 Source Select Register 2	
58h	CHECKEVT3SEL3	Checker X-Bar EVT3 Source Select Register 3	
60h	CHECKEVT4SEL1	Checker X-Bar EVT4 Source Select Register 1	
64h	CHECKEVT4SEL2	Checker X-Bar EVT4 Source Select Register 2	
68h	CHECKEVT4SEL3	Checker X-Bar EVT4 Source Select Register 3	

Complex bit access types are encoded to fit into small table cells. Table 24-238 shows the codes that are used for access types in this section.

Table 24-238. ADC_SAFECHECK_INTEVT_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		

Table 24-238. ADC_SAFECHECK_INTEVT_REGS Access Type Codes (continued)

Access Type	Code	Description
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

24.16.5.1 OOTFLG Register (Offset = 0h) [Reset = 00000000h]

OOTFLG is shown in Figure 24-257 and described in Table 24-239.

Return to the Summary Table.

Checker Out-of-Tolerance Flag Register

Figure 24-257. OOTFLG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
OOT16	OOT15	OOT14	OOT13	OOT12	OOT11	OOT10	OOT9
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
OOT8	OOT7	OOT6	OOT5	OOT4	OOT3	OOT2	OOT1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 24-239. OOTFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	OOT16	R	0h	ADC results safety checker 16 out-of-tolerance flag. Set when CHECK16 detects a difference between configured conversion results greater than the configured tolerance. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
14	OOT15	R	0h	ADC results safety checker 15 out-of-tolerance flag. Set when CHECK15 detects a difference between configured conversion results greater than the configured tolerance. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
13	OOT14	R	0h	ADC results safety checker 14 out-of-tolerance flag. Set when CHECK14 detects a difference between configured conversion results greater than the configured tolerance. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
12	OOT13	R	0h	ADC results safety checker 13 out-of-tolerance flag. Set when CHECK13 detects a difference between configured conversion results greater than the configured tolerance. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn

Table 24-239. OOTFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	OOT12	R	0h	ADC results safety checker 12 out-of-tolerance flag. Set when CHECK12 detects a difference between configured conversion results greater than the configured tolerance. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
10	OOT11	R	0h	ADC results safety checker 11 out-of-tolerance flag. Set when CHECK11 detects a difference between configured conversion results greater than the configured tolerance. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
9	OOT10	R	0h	ADC results safety checker 10 out-of-tolerance flag. Set when CHECK10 detects a difference between configured conversion results greater than the configured tolerance. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
8	OOT9	R	0h	ADC results safety checker 1 out-of-tolerance flag. Set when CHECK9 detects a difference between configured conversion results greater than the configured tolerance. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
7	OOT8	R	0h	ADC results safety checker 8 out-of-tolerance flag. Set when CHECK8 detects a difference between configured conversion results greater than the configured tolerance. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
6	OOT7	R	0h	ADC results safety checker 7 out-of-tolerance flag. Set when CHECK7 detects a difference between configured conversion results greater than the configured tolerance. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
5	OOT6	R	0h	ADC results safety checker 6 out-of-tolerance flag. Set when CHECK6 detects a difference between configured conversion results greater than the configured tolerance. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn

Table 24-239. OOTFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	OOT5	R	0h	ADC results safety checker 5 out-of-tolerance flag. Set when CHECK5 detects a difference between configured conversion results greater than the configured tolerance. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
3	OOT4	R	0h	ADC results safety checker 4 out-of-tolerance flag. Set when CHECK4 detects a difference between configured conversion results greater than the configured tolerance. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
2	OOT3	R	0h	ADC results safety checker 3 out-of-tolerance flag. Set when CHECK3 detects a difference between configured conversion results greater than the configured tolerance. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
1	OOT2	R	0h	ADC results safety checker 2 out-of-tolerance flag. Set when CHECK2 detects a difference between configured conversion results greater than the configured tolerance. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
0	OOT1	R	0h	ADC results safety checker 1 out-of-tolerance flag. Set when CHECK1 detects a difference between configured conversion results greater than the configured tolerance. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn

24.16.5.2 OOTFLGCLR Register (Offset = 4h) [Reset = 00000000h]

OOTFLGCLR is shown in Figure 24-258 and described in Table 24-240.

Return to the Summary Table.

Checker Out-of-Tolerance Flag Clear Register

Figure 24-258. OOTFLGCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
OOT16	OOT15	OOT14	OOT13	OOT12	OOT11	OOT10	OOT9
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
OOT8	OOT7	OOT6	OOT5	OOT4	OOT3	OOT2	OOT1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 24-240. OOTFLGCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	OOT16	R-0/W1S	0h	ADC results safety checker 16 out-of-tolerance flag clear. Used to clear OOT status from CHECK16. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
14	OOT15	R-0/W1S	0h	ADC results safety checker 15 out-of-tolerance flag clear. Used to clear OOT status from CHECK15. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
13	OOT14	R-0/W1S	0h	ADC results safety checker 14 out-of-tolerance flag clear. Used to clear OOT status from CHECK14. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
12	OOT13	R-0/W1S	0h	ADC results safety checker 135 out-of-tolerance flag clear. Used to clear OOT status from CHECK13. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn

Table 24-240. OOTFLGCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	OOT12	R-0/W1S	0h	ADC results safety checker 12 out-of-tolerance flag clear. Used to clear OOT status from CHECK12. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
10	OOT11	R-0/W1S	0h	ADC results safety checker 11 out-of-tolerance flag clear. Used to clear OOT status from CHECK11. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
9	OOT10	R-0/W1S	0h	ADC results safety checker 10 out-of-tolerance flag clear. Used to clear OOT status from CHECK10. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
8	OOT9	R-0/W1S	0h	ADC results safety checker 9 out-of-tolerance flag clear. Used to clear OOT status from CHECK9. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
7	OOT8	R-0/W1S	0h	ADC results safety checker 8 out-of-tolerance flag clear. Used to clear OOT status from CHECK8. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
6	OOT7	R-0/W1S	0h	ADC results safety checker 7 out-of-tolerance flag clear. Used to clear OOT status from CHECK7. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
5	OOT6	R-0/W1S	0h	ADC results safety checker 6 out-of-tolerance flag clear. Used to clear OOT status from CHECK6. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn

Table 24-240. OOTFLGCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	OOT5	R-0/W1S	0h	ADC results safety checker 5 out-of-tolerance flag clear. Used to clear OOT status from CHECK5. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
3	OOT4	R-0/W1S	0h	ADC results safety checker 4 out-of-tolerance flag clear. Used to clear OOT status from CHECK4. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
2	OOT3	R-0/W1S	0h	ADC results safety checker 3 out-of-tolerance flag clear. Used to clear OOT status from CHECK3. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
1	OOT2	R-0/W1S	0h	ADC results safety checker 2 out-of-tolerance flag clear. Used to clear OOT status from CHECK2. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
0	OOT1	R-0/W1S	0h	ADC results safety checker 1 out-of-tolerance flag clear. Used to clear OOT status from CHECK1. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn

24.16.5.3 RES1OVF Register (Offset = 8h) [Reset = 00000000h]

RES1OVF is shown in Figure 24-259 and described in Table 24-241.

Return to the Summary Table.

Checker Overflow Result 1 Flag Register

Figure 24-259. RES1OVF Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RES1OVF16	RES1OVF15	RES1OVF14	RES1OVF13	RES1OVF12	RES1OVF11	RES1OVF10	RES1OVF9
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RES1OVF8	RES1OVF7	RES1OVF6	RES1OVF5	RES1OVF4	RES1OVF3	RES1OVF2	RES1OVF1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 24-241. RES1OVF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RES1OVF16	R	0h	ADC results safety checker 16 overflow flag for result 1. Set when CHECK16 detects that conversion result 1 has arrived more than once before result 2 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
14	RES1OVF15	R	0h	ADC results safety checker 15 overflow flag for result 1. Set when CHECK15 detects that conversion result 1 has arrived more than once before result 2 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
13	RES1OVF14	R	0h	ADC results safety checker 14 overflow flag for result 1. Set when CHECK14 detects that conversion result 1 has arrived more than once before result 2 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
12	RES1OVF13	R	0h	ADC results safety checker 13 overflow flag for result 1. Set when CHECK13 detects that conversion result 1 has arrived more than once before result 2 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn

Table 24-241. RES1OVF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	RES1OVF12	R	0h	ADC results safety checker 12 overflow flag for result 1. Set when CHECK12 detects that conversion result 1 has arrived more than once before result 2 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
10	RES1OVF11	R	0h	ADC results safety checker 11 overflow flag for result 1. Set when CHECK11 detects that conversion result 1 has arrived more than once before result 2 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
9	RES1OVF10	R	0h	ADC results safety checker 10 overflow flag for result 1. Set when CHECK10 detects that conversion result 1 has arrived more than once before result 2 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
8	RES1OVF9	R	0h	ADC results safety checker 9 overflow flag for result 1. Set when CHECK9 detects that conversion result 1 has arrived more than once before result 2 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
7	RES1OVF8	R	0h	ADC results safety checker 8 overflow flag for result 1. Set when CHECK8 detects that conversion result 1 has arrived more than once before result 2 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
6	RES1OVF7	R	0h	ADC results safety checker 7 overflow flag for result 1. Set when CHECK7 detects that conversion result 1 has arrived more than once before result 2 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
5	RES1OVF6	R	0h	ADC results safety checker 6 overflow flag for result 1. Set when CHECK6 detects that conversion result 1 has arrived more than once before result 2 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn

Table 24-241. RES1OVF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	RES1OVF5	R	0h	ADC results safety checker 5 overflow flag for result 1. Set when CHECK5 detects that conversion result 1 has arrived more than once before result 2 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
3	RES1OVF4	R	0h	ADC results safety checker 4 overflow flag for result 1. Set when CHECK4 detects that conversion result 1 has arrived more than once before result 2 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
2	RES1OVF3	R	0h	ADC results safety checker 3 overflow flag for result 1. Set when CHECK3 detects that conversion result 1 has arrived more than once before result 2 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
1	RES1OVF2	R	0h	ADC results safety checker 2 overflow flag for result 1. Set when CHECK2 detects that conversion result 1 has arrived more than once before result 2 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
0	RES1OVF1	R	0h	ADC results safety checker 1 overflow flag for result 1. Set when CHECK1 detects that conversion result 1 has arrived more than once before result 2 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn

24.16.5.4 RES1OVFCLR Register (Offset = Ch) [Reset = 00000000h]

RES1OVFCLR is shown in Figure 24-260 and described in Table 24-242.

Return to the Summary Table.

Checker Overflow Result 1 Flag Clear Register

Figure 24-260. RES1OVFCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RES1OVF16	RES1OVF15	RES1OVF14	RES1OVF13	RES1OVF12	RES1OVF11	RES1OVF10	RES1OVF9
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RES1OVF8	RES1OVF7	RES1OVF6	RES1OVF5	RES1OVF4	RES1OVF3	RES1OVF2	RES1OVF1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 24-242. RES1OVFCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RES1OVF16	R-0/W1S	0h	ADC results safety checker 16 result 1 overflow flag clear. Used to clear RES1OVF status from CHECK16. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
14	RES1OVF15	R-0/W1S	0h	ADC results safety checker 15 result 1 overflow flag clear. Used to clear RES1OVF status from CHECK15. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
13	RES1OVF14	R-0/W1S	0h	ADC results safety checker 14 result 1 overflow flag clear. Used to clear RES1OVF status from CHECK14. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
12	RES1OVF13	R-0/W1S	0h	ADC results safety checker 13 result 1 overflow flag clear. Used to clear RES1OVF status from CHECK13. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn

Table 24-242. RES1OVFCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	RES1OVF12	R-0/W1S	0h	ADC results safety checker 12 result 1 overflow flag clear. Used to clear RES1OVF status from CHECK12. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
10	RES1OVF11	R-0/W1S	0h	ADC results safety checker 11 result 1 overflow flag clear. Used to clear RES1OVF status from CHECK11. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
9	RES1OVF10	R-0/W1S	0h	ADC results safety checker 10 result 1 overflow flag clear. Used to clear RES1OVF status from CHECK10. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
8	RES1OVF9	R-0/W1S	0h	ADC results safety checker 9 result 1 overflow flag clear. Used to clear RES1OVF status from CHECK9. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
7	RES1OVF8	R-0/W1S	0h	ADC results safety checker 8 result 1 overflow flag clear. Used to clear RES1OVF status from CHECK8. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
6	RES1OVF7	R-0/W1S	0h	ADC results safety checker 7 result 1 overflow flag clear. Used to clear RES1OVF status from CHECK7. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
5	RES1OVF6	R-0/W1S	0h	ADC results safety checker 6 result 1 overflow flag clear. Used to clear RES1OVF status from CHECK6. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn

Table 24-242. RES1OVFCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	RES1OVF5	R-0/W1S	0h	ADC results safety checker 5 result 1 overflow flag clear. Used to clear RES1OVF status from CHECK5. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
3	RES1OVF4	R-0/W1S	0h	ADC results safety checker 4 result 1 overflow flag clear. Used to clear RES1OVF status from CHECK4. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
2	RES1OVF3	R-0/W1S	0h	ADC results safety checker 3 result 1 overflow flag clear. Used to clear RES1OVF status from CHECK3. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
1	RES1OVF2	R-0/W1S	0h	ADC results safety checker 2 result 1 overflow flag clear. Used to clear RES1OVF status from CHECK2. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
0	RES1OVF1	R-0/W1S	0h	ADC results safety checker 1 result 1 overflow flag clear. Used to clear RES1OVF status from CHECK1. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn

24.16.5.5 RES2OVF Register (Offset = 10h) [Reset = 00000000h]

RES2OVF is shown in Figure 24-261 and described in Table 24-243.

Return to the Summary Table.

Checker Overflow Result 2 Flag Register

Figure 24-261. RES2OVF Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RES2OVF16	RES2OVF15	RES2OVF14	RES2OVF13	RES2OVF12	RES2OVF11	RES2OVF10	RES2OVF9
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RES2OVF8	RES2OVF7	RES2OVF6	RES2OVF5	RES2OVF4	RES2OVF3	RES2OVF2	RES2OVF1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 24-243. RES2OVF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RES2OVF16	R	0h	ADC results safety checker 16 overflow flag for result 2. Set when CHECK16 detects that conversion result 2 has arrived more than once before result 1 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
14	RES2OVF15	R	0h	ADC results safety checker 15 overflow flag for result 2. Set when CHECK15 detects that conversion result 2 has arrived more than once before result 1 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
13	RES2OVF14	R	0h	ADC results safety checker 14 overflow flag for result 2. Set when CHECK14 detects that conversion result 2 has arrived more than once before result 1 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
12	RES2OVF13	R	0h	ADC results safety checker 13 overflow flag for result 2. Set when CHECK13 detects that conversion result 2 has arrived more than once before result 1 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn

Table 24-243. RES2OVF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	RES2OVF12	R	0h	ADC results safety checker 12 overflow flag for result 2. Set when CHECK12 detects that conversion result 2 has arrived more than once before result 1 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
10	RES2OVF11	R	0h	ADC results safety checker 11 overflow flag for result 2. Set when CHECK11 detects that conversion result 2 has arrived more than once before result 1 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
9	RES2OVF10	R	0h	ADC results safety checker 10 overflow flag for result 2. Set when CHECK10 detects that conversion result 2 has arrived more than once before result 1 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
8	RES2OVF9	R	0h	ADC results safety checker 9 overflow flag for result 2. Set when CHECK9 detects that conversion result 2 has arrived more than once before result 1 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
7	RES2OVF8	R	0h	ADC results safety checker 8 overflow flag for result 2. Set when CHECK8 detects that conversion result 2 has arrived more than once before result 1 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
6	RES2OVF7	R	0h	ADC results safety checker 7 overflow flag for result 2. Set when CHECK7 detects that conversion result 2 has arrived more than once before result 1 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
5	RES2OVF6	R	0h	ADC results safety checker 6 overflow flag for result 2. Set when CHECK6 detects that conversion result 2 has arrived more than once before result 1 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn

Table 24-243. RES2OVF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	RES2OVF5	R	0h	ADC results safety checker 5 overflow flag for result 2. Set when CHECK5 detects that conversion result 2 has arrived more than once before result 1 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
3	RES2OVF4	R	0h	ADC results safety checker 4 overflow flag for result 2. Set when CHECK4 detects that conversion result 2 has arrived more than once before result 1 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
2	RES2OVF3	R	0h	ADC results safety checker 3 overflow flag for result 2. Set when CHECK3 detects that conversion result 2 has arrived more than once before result 1 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
1	RES2OVF2	R	0h	ADC results safety checker 2 overflow flag for result 2. Set when CHECK2 detects that conversion result 2 has arrived more than once before result 1 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn
0	RES2OVF1	R	0h	ADC results safety checker 1 overflow flag for result 2. Set when CHECK1 detects that conversion result 2 has arrived more than once before result 1 has arrived. Use the CHECKINTSEL1, CHECKINTSEL2, CHECKINTSEL3, CHECKEVTxSEL1, CHECKEVTxSEL2, and CHECKEVTxSEL3 registers to aggregate detected OOT and OVF flags into interrupt (INT) or X-bar events (EVT). Reset type: SYSRSn

24.16.5.6 RES2OVFCLR Register (Offset = 14h) [Reset = 00000000h]

RES2OVFCLR is shown in Figure 24-262 and described in Table 24-244.

Return to the Summary Table.

Checker Overflow Result 2 Flag Clear Register

Figure 24-262. RES2OVFCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RES2OVF16	RES2OVF15	RES2OVF14	RES2OVF13	RES2OVF12	RES2OVF11	RES2OVF10	RES2OVF9
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RES2OVF8	RES2OVF7	RES2OVF6	RES2OVF5	RES2OVF4	RES2OVF3	RES2OVF2	RES2OVF1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 24-244. RES2OVFCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RES2OVF16	R-0/W1S	0h	ADC results safety checker 16 result overflow flag clear. Used to clear OVF status from CHECK16. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHECKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
14	RES2OVF15	R-0/W1S	0h	ADC results safety checker 15 result overflow flag clear. Used to clear OVF status from CHECK15. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHECKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
13	RES2OVF14	R-0/W1S	0h	ADC results safety checker 14 result overflow flag clear. Used to clear OVF status from CHECK14. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHECKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
12	RES2OVF13	R-0/W1S	0h	ADC results safety checker 13 result overflow flag clear. Used to clear OVF status from CHECK13. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn

Table 24-244. RES2OVFCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	RES2OVF12	R-0/W1S	0h	ADC results safety checker 12 result overflow flag clear. Used to clear OVF status from CHECK12. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHECKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
10	RES2OVF11	R-0/W1S	0h	ADC results safety checker 11 result overflow flag clear. Used to clear OVF status from CHECK11. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHECKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
9	RES2OVF10	R-0/W1S	0h	ADC results safety checker 10 result overflow flag clear. Used to clear OVF status from CHECK10. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHECKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
8	RES2OVF9	R-0/W1S	0h	ADC results safety checker 9 result overflow flag clear. Used to clear OVF status from CHECK9. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
7	RES2OVF8	R-0/W1S	0h	ADC results safety checker 8 result overflow flag clear. Used to clear OVF status from CHECK8. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHECKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
6	RES2OVF7	R-0/W1S	0h	ADC results safety checker 7 result overflow flag clear. Used to clear OVF status from CHECK7. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHECKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
5	RES2OVF6	R-0/W1S	0h	ADC results safety checker 6 result overflow flag clear. Used to clear OVF status from CHECK6. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHECKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn

Table 24-244. RES2OVFCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	RES2OVF5	R-0/W1S	0h	ADC results safety checker 5 result overflow flag clear. Used to clear OVF status from CHECK5. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
3	RES2OVF4	R-0/W1S	0h	ADC results safety checker 4 result overflow flag clear. Used to clear OVF status from CHECK4. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHECKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
2	RES2OVF3	R-0/W1S	0h	ADC results safety checker 3 result overflow flag clear. Used to clear OVF status from CHECK3. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHECKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
1	RES2OVF2	R-0/W1S	0h	ADC results safety checker 2 result overflow flag clear. Used to clear OVF status from CHECK2. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHECKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn
0	RES2OVF1	R-0/W1S	0h	ADC results safety checker 1 result overflow flag clear. Used to clear OVF status from CHECK1. In the case of a safety checker interrupt, clear all serviced OOT or OVF flags first, then clear the global interrupt flag (in the CHECKINTFLG register) using the CHKINTFLGCLR register. In the case of a safety checker X-bar event, clear all associated OOT or OVF flags to clear the event. Reset type: SYSRSn

24.16.5.7 CHECKINTFLG Register (Offset = 18h) [Reset = 0000h]

CHECKINTFLG is shown in Figure 24-263 and described in Table 24-245.

Return to the Summary Table.

Checker Interrupt Flag Register

Figure 24-263. CHECKINTFLG Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CHECKINT
R-0h							R-0h

Table 24-245. CHECKINTFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R	0h	Reserved
0	CHECKINT	R	0h	ADC results safety checker subsytem interrupt flag. Indicates that one or more configured OOT or OVF conditions have occurred in the individual safety checker modules. In the ISR, clear all serviced OOT or OVF flags first (using the OOTFLGCLR and OVFFLGCLR registers), then clear this flag using the CHKINTFLGCLR register. The CHECKINTSEL1 and CHECKINTSEL2 registers are used to select which OOT and OVF flags from the individual checker modules can trigger this interrupt. Reset type: SYSRSn

24.16.5.8 CHECKINTFLGCLR Register (Offset = 1Ch) [Reset = 0000h]

CHECKINTFLGCLR is shown in Figure 24-264 and described in Table 24-246.

Return to the Summary Table.

Checker Interrupt Flag Clear Register

Figure 24-264. CHECKINTFLGCLR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CHECKINTCLR
R-0h							R-0/W1S-0h

Table 24-246. CHECKINTFLGCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R	0h	Reserved
0	CHECKINTCLR	R-0/W1S	0h	ADC results safety checker subsytem interrupt flag clear. Used to clear the global safety checker subsystem interrupt flag. In the ISR, clear all serviced OOT or OVF flags first (using the OOTFLGCLR and OVFFLGCLR registers), then clear the global CHECKINT flag using the this register. Reset type: SYSRSn

24.16.5.9 CHECKINTSEL1 Register (Offset = 20h) [Reset = 00000000h]

CHECKINTSEL1 is shown in Figure 24-265 and described in Table 24-247.

Return to the Summary Table.

Checker Interrupt Source Select Register 1

Figure 24-265. CHECKINTSEL1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RES1OVF16EN	RES1OVF15EN	RES1OVF14EN	RES1OVF13EN	RES1OVF12EN	RES1OVF11EN	RES1OVF10EN	RES1OVF9EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RES1OVF8EN	RES1OVF7EN	RES1OVF6EN	RES1OVF5EN	RES1OVF4EN	RES1OVF3EN	RES1OVF2EN	RES1OVF1EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-247. CHECKINTSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RES1OVF16EN	R/W	0h	Enable CHECK16 RES1OVF as a source for CHECKINT. Reset type: SYSRSn
14	RES1OVF15EN	R/W	0h	Enable CHECK15 RES1OVF as a source for CHECKINT. Reset type: SYSRSn
13	RES1OVF14EN	R/W	0h	Enable CHECK14 RES1OVF as a source for CHECKINT. Reset type: SYSRSn
12	RES1OVF13EN	R/W	0h	Enable CHECK13 RES1OVF as a source for CHECKINT. Reset type: SYSRSn
11	RES1OVF12EN	R/W	0h	Enable CHECK12 RES1OVF as a source for CHECKINT. Reset type: SYSRSn
10	RES1OVF11EN	R/W	0h	Enable CHECK11 RES1OVF as a source for CHECKINT. Reset type: SYSRSn
9	RES1OVF10EN	R/W	0h	Enable CHECK10 RES1OVF as a source for CHECKINT. Reset type: SYSRSn
8	RES1OVF9EN	R/W	0h	Enable CHECK9 RES1OVF as a source for CHECKINT. Reset type: SYSRSn
7	RES1OVF8EN	R/W	0h	Enable CHECK8 RES1OVF as a source for CHECKINT. Reset type: SYSRSn
6	RES1OVF7EN	R/W	0h	Enable CHECK7 RES1OVF as a source for CHECKINT. Reset type: SYSRSn
5	RES1OVF6EN	R/W	0h	Enable CHECK6 RES1OVF as a source for CHECKINT. Reset type: SYSRSn
4	RES1OVF5EN	R/W	0h	Enable CHECK5 RES1OVF as a source for CHECKINT. Reset type: SYSRSn
3	RES1OVF4EN	R/W	0h	Enable CHECK4 RES1OVF as a source for CHECKINT. Reset type: SYSRSn

Table 24-247. CHECKINTSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RES1OVF3EN	R/W	0h	Enable CHECK3 RES1OVF as a source for CHECKINT. Reset type: SYSRSn
1	RES1OVF2EN	R/W	0h	Enable CHECK2 RES1OVF as a source for CHECKINT. Reset type: SYSRSn
0	RES1OVF1EN	R/W	0h	Enable CHECK1 RES1OVF as a source for CHECKINT. Reset type: SYSRSn

24.16.5.10 CHECKINTSEL2 Register (Offset = 24h) [Reset = 00000000h]

CHECKINTSEL2 is shown in Figure 24-266 and described in Table 24-248.

Return to the Summary Table.

Checker Interrupt Source Select Register 2

Figure 24-266. CHECKINTSEL2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RES2OVF16EN	RES2OVF15EN	RES2OVF14EN	RES2OVF13EN	RES2OVF12EN	RES2OVF11EN	RES2OVF10EN	RES2OVF9EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RES2OVF8EN	RES2OVF7EN	RES2OVF6EN	RES2OVF5EN	RES2OVF4EN	RES2OVF3EN	RES2OVF2EN	RES2OVF1EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-248. CHECKINTSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RES2OVF16EN	R/W	0h	Enable CHECK16 RES2OVF as a source for CHECKINT. Reset type: SYSRSn
14	RES2OVF15EN	R/W	0h	Enable CHECK15 RES2OVF as a source for CHECKINT. Reset type: SYSRSn
13	RES2OVF14EN	R/W	0h	Enable CHECK14 RES2OVF as a source for CHECKINT. Reset type: SYSRSn
12	RES2OVF13EN	R/W	0h	Enable CHECK13 RES2OVF as a source for CHECKINT. Reset type: SYSRSn
11	RES2OVF12EN	R/W	0h	Enable CHECK12 RES2OVF as a source for CHECKINT. Reset type: SYSRSn
10	RES2OVF11EN	R/W	0h	Enable CHECK11 RES2OVF as a source for CHECKINT. Reset type: SYSRSn
9	RES2OVF10EN	R/W	0h	Enable CHECK10 RES2OVF as a source for CHECKINT. Reset type: SYSRSn
8	RES2OVF9EN	R/W	0h	Enable CHECK9 RES2OVF as a source for CHECKINT. Reset type: SYSRSn
7	RES2OVF8EN	R/W	0h	Enable CHECK8 RES2OVF as a source for CHECKINT. Reset type: SYSRSn
6	RES2OVF7EN	R/W	0h	Enable CHECK7 RES2OVF as a source for CHECKINT. Reset type: SYSRSn
5	RES2OVF6EN	R/W	0h	Enable CHECK6 RES2OVF as a source for CHECKINT. Reset type: SYSRSn
4	RES2OVF5EN	R/W	0h	Enable CHECK5 RES2OVF as a source for CHECKINT. Reset type: SYSRSn
3	RES2OVF4EN	R/W	0h	Enable CHECK4 RES2OVF as a source for CHECKINT. Reset type: SYSRSn

Table 24-248. CHECKINTSEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RES2OVF3EN	R/W	0h	Enable CHECK3 RES2OVF as a source for CHECKINT. Reset type: SYSRSn
1	RES2OVF2EN	R/W	0h	Enable CHECK2 RES2OVF as a source for CHECKINT. Reset type: SYSRSn
0	RES2OVF1EN	R/W	0h	Enable CHECK1 RES2OVF as a source for CHECKINT. Reset type: SYSRSn

24.16.5.11 CHECKINTSEL3 Register (Offset = 28h) [Reset = 00000000h]

CHECKINTSEL3 is shown in Figure 24-267 and described in Table 24-249.

Return to the Summary Table.

Checker Interrupt Source Select Register 3

Figure 24-267. CHECKINTSEL3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
OOT16EN	OOT15EN	OOT14EN	OOT13EN	OOT12EN	OOT11EN	OOT10EN	OOT9EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
OOT8EN	OOT7EN	OOT6EN	OOT5EN	OOT4EN	OOT3EN	OOT2EN	OOT1EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-249. CHECKINTSEL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	OOT16EN	R/W	0h	Enable CHECK16 OOT as a source for CHECKINT. Reset type: SYSRSn
14	OOT15EN	R/W	0h	Enable CHECK15 OOT as a source for CHECKINT. Reset type: SYSRSn
13	OOT14EN	R/W	0h	Enable CHECK14 OOT as a source for CHECKINT. Reset type: SYSRSn
12	OOT13EN	R/W	0h	Enable CHECK13 OOT as a source for CHECKINT. Reset type: SYSRSn
11	OOT12EN	R/W	0h	Enable CHECK12 OOT as a source for CHECKINT. Reset type: SYSRSn
10	OOT11EN	R/W	0h	Enable CHECK11 OOT as a source for CHECKINT. Reset type: SYSRSn
9	OOT10EN	R/W	0h	Enable CHECK10 OOT as a source for CHECKINT. Reset type: SYSRSn
8	OOT9EN	R/W	0h	Enable CHECK9 OOT as a source for CHECKINT. Reset type: SYSRSn
7	OOT8EN	R/W	0h	Enable CHECK8 OOT as a source for CHECKINT. Reset type: SYSRSn
6	OOT7EN	R/W	0h	Enable CHECK7 OOT as a source for CHECKINT. Reset type: SYSRSn
5	OOT6EN	R/W	0h	Enable CHECK6 OOT as a source for CHECKINT. Reset type: SYSRSn
4	OOT5EN	R/W	0h	Enable CHECK5 OOT as a source for CHECKINT. Reset type: SYSRSn
3	OOT4EN	R/W	0h	Enable CHECK4 OOT as a source for CHECKINT. Reset type: SYSRSn

Table 24-249. CHECKINTSEL3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	OOT3EN	R/W	0h	Enable CHECK3 OOT as a source for CHECKINT. Reset type: SYSRSn
1	OOT2EN	R/W	0h	Enable CHECK2 OOT as a source for CHECKINT. Reset type: SYSRSn
0	OOT1EN	R/W	0h	Enable CHECK1 OOT as a source for CHECKINT. Reset type: SYSRSn

24.16.5.12 CHECKEVT1SEL1 Register (Offset = 30h) [Reset = 00000000h]

CHECKEVT1SEL1 is shown in Figure 24-268 and described in Table 24-250.

Return to the Summary Table.

Checker X-Bar EVT1 Source Select Register 1

Figure 24-268. CHECKEVT1SEL1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RES1OVF16EN	RES1OVF15EN	RES1OVF14EN	RES1OVF13EN	RES1OVF12EN	RES1OVF11EN	RES1OVF10EN	RES1OVF9EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RES1OVF8EN	RES1OVF7EN	RES1OVF6EN	RES1OVF5EN	RES1OVF4EN	RES1OVF3EN	RES1OVF2EN	RES1OVF1EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-250. CHECKEVT1SEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RES1OVF16EN	R/W	0h	Enable CHECK16 RES1OVF as a source for CHECKEVT1. Reset type: SYSRSn
14	RES1OVF15EN	R/W	0h	Enable CHECK15 RES1OVF as a source for CHECKEVT1. Reset type: SYSRSn
13	RES1OVF14EN	R/W	0h	Enable CHECK14 RES1OVF as a source for CHECKEVT1. Reset type: SYSRSn
12	RES1OVF13EN	R/W	0h	Enable CHECK13 RES1OVF as a source for CHECKEVT1. Reset type: SYSRSn
11	RES1OVF12EN	R/W	0h	Enable CHECK12 RES1OVF as a source for CHECKEVT1. Reset type: SYSRSn
10	RES1OVF11EN	R/W	0h	Enable CHECK11 RES1OVF as a source for CHECKEVT1. Reset type: SYSRSn
9	RES1OVF10EN	R/W	0h	Enable CHECK10 RES1OVF as a source for CHECKEVT1. Reset type: SYSRSn
8	RES1OVF9EN	R/W	0h	Enable CHECK9 RES1OVF as a source for CHECKEVT1. Reset type: SYSRSn
7	RES1OVF8EN	R/W	0h	Enable CHECK8 RES1OVF as a source for CHECKEVT1. Reset type: SYSRSn
6	RES1OVF7EN	R/W	0h	Enable CHECK7 RES1OVF as a source for CHECKEVT1. Reset type: SYSRSn
5	RES1OVF6EN	R/W	0h	Enable CHECK6 RES1OVF as a source for CHECKEVT1. Reset type: SYSRSn
4	RES1OVF5EN	R/W	0h	Enable CHECK5 RES1OVF as a source for CHECKEVT1. Reset type: SYSRSn
3	RES1OVF4EN	R/W	0h	Enable CHECK4 RES1OVF as a source for CHECKEVT1. Reset type: SYSRSn

Table 24-250. CHECKEVT1SEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RES1OVF3EN	R/W	0h	Enable CHECK3 RES1OVF as a source for CHECKEVT1. Reset type: SYSRSn
1	RES1OVF2EN	R/W	0h	Enable CHECK2 RES1OVF as a source for CHECKEVT1. Reset type: SYSRSn
0	RES1OVF1EN	R/W	0h	Enable CHECK1 RES1OVF as a source for CHECKEVT1. Reset type: SYSRSn

24.16.5.13 CHECKEVT1SEL2 Register (Offset = 34h) [Reset = 00000000h]

CHECKEVT1SEL2 is shown in Figure 24-269 and described in Table 24-251.

Return to the Summary Table.

Checker X-Bar EVT1 Source Select Register 2

Figure 24-269. CHECKEVT1SEL2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RES2OVF16EN	RES2OVF15EN	RES2OVF14EN	RES2OVF13EN	RES2OVF12EN	RES2OVF11EN	RES2OVF10EN	RES2OVF9EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RES2OVF8EN	RES2OVF7EN	RES2OVF6EN	RES2OVF5EN	RES2OVF4EN	RES2OVF3EN	RES2OVF2EN	RES2OVF1EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-251. CHECKEVT1SEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RES2OVF16EN	R/W	0h	Enable CHECK16 RES2OVF as a source for CHECKEVT1. Reset type: SYSRSn
14	RES2OVF15EN	R/W	0h	Enable CHECK15 RES2OVF as a source for CHECKEVT1. Reset type: SYSRSn
13	RES2OVF14EN	R/W	0h	Enable CHECK14 RES2OVF as a source for CHECKEVT1. Reset type: SYSRSn
12	RES2OVF13EN	R/W	0h	Enable CHECK13 RES2OVF as a source for CHECKEVT1. Reset type: SYSRSn
11	RES2OVF12EN	R/W	0h	Enable CHECK12 RES2OVF as a source for CHECKEVT1. Reset type: SYSRSn
10	RES2OVF11EN	R/W	0h	Enable CHECK11 RES2OVF as a source for CHECKEVT1. Reset type: SYSRSn
9	RES2OVF10EN	R/W	0h	Enable CHECK10 RES2OVF as a source for CHECKEVT1. Reset type: SYSRSn
8	RES2OVF9EN	R/W	0h	Enable CHECK9 RES2OVF as a source for CHECKEVT1. Reset type: SYSRSn
7	RES2OVF8EN	R/W	0h	Enable CHECK8 RES2OVF as a source for CHECKEVT1. Reset type: SYSRSn
6	RES2OVF7EN	R/W	0h	Enable CHECK7 RES2OVF as a source for CHECKEVT1. Reset type: SYSRSn
5	RES2OVF6EN	R/W	0h	Enable CHECK6 RES2OVF as a source for CHECKEVT1. Reset type: SYSRSn
4	RES2OVF5EN	R/W	0h	Enable CHECK5 RES2OVF as a source for CHECKEVT1. Reset type: SYSRSn
3	RES2OVF4EN	R/W	0h	Enable CHECK4 RES2OVF as a source for CHECKEVT1. Reset type: SYSRSn

Table 24-251. CHECKEVT1SEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RES2OVF3EN	R/W	0h	Enable CHECK3 RES2OVF as a source for CHECKEVT1. Reset type: SYSRSn
1	RES2OVF2EN	R/W	0h	Enable CHECK2 RES2OVF as a source for CHECKEVT1. Reset type: SYSRSn
0	RES2OVF1EN	R/W	0h	Enable CHECK1 RES2OVF as a source for CHECKEVT1. Reset type: SYSRSn

24.16.5.14 CHECKEVT1SEL3 Register (Offset = 38h) [Reset = 00000000h]

CHECKEVT1SEL3 is shown in Figure 24-270 and described in Table 24-252.

Return to the Summary Table.

Checker X-Bar EVT1 Source Select Register 3

Figure 24-270. CHECKEVT1SEL3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
OOT16EN	OOT15EN	OOT14EN	OOT13EN	OOT12EN	OOT11EN	OOT10EN	OOT9EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
OOT8EN	OOT7EN	OOT6EN	OOT5EN	OOT4EN	OOT3EN	OOT2EN	OOT1EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-252. CHECKEVT1SEL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	OOT16EN	R/W	0h	Enable CHECK16 OOT as a source for CHECKEVT1. Reset type: SYSRSn
14	OOT15EN	R/W	0h	Enable CHECK15 OOTas a source for CHECKEVT1. Reset type: SYSRSn
13	OOT14EN	R/W	0h	Enable CHECK14 OOT as a source for CHECKEVT1. Reset type: SYSRSn
12	OOT13EN	R/W	0h	Enable CHECK13 OOT as a source for CHECKEVT1. Reset type: SYSRSn
11	OOT12EN	R/W	0h	Enable CHECK12 OOT as a source for CHECKEVT1. Reset type: SYSRSn
10	OOT11EN	R/W	0h	Enable CHECK11 OOTas a source for CHECKEVT1. Reset type: SYSRSn
9	OOT10EN	R/W	0h	Enable CHECK10 OOT as a source for CHECKEVT1. Reset type: SYSRSn
8	OOT9EN	R/W	0h	Enable CHECK9 OOT as a source for CHECKEVT1. Reset type: SYSRSn
7	OOT8EN	R/W	0h	Enable CHECK8 OOT as a source for CHECKEVT1. Reset type: SYSRSn
6	OOT7EN	R/W	0h	Enable CHECK7 OOTas a source for CHECKEVT1. Reset type: SYSRSn
5	OOT6EN	R/W	0h	Enable CHECK6 OOT as a source for CHECKEVT1. Reset type: SYSRSn
4	OOT5EN	R/W	0h	Enable CHECK5 OOT as a source for CHECKEVT1. Reset type: SYSRSn
3	OOT4EN	R/W	0h	Enable CHECK4 OOT as a source for CHECKEVT1. Reset type: SYSRSn

Table 24-252. CHECKEVT1SEL3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	OOT3EN	R/W	0h	Enable CHECK3 OOTas a source for CHECKEVT1. Reset type: SYSRSn
1	OOT2EN	R/W	0h	Enable CHECK2 OOT as a source for CHECKEVT1. Reset type: SYSRSn
0	OOT1EN	R/W	0h	Enable CHECK1 OOT as a source for CHECKEVT1. Reset type: SYSRSn

24.16.5.15 CHECKEVT2SEL1 Register (Offset = 40h) [Reset = 00000000h]

CHECKEVT2SEL1 is shown in Figure 24-271 and described in Table 24-253.

Return to the Summary Table.

Checker X-Bar EVT2 Source Select Register 1

Figure 24-271. CHECKEVT2SEL1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RES1OVF16EN	RES1OVF15EN	RES1OVF14EN	RES1OVF13EN	RES1OVF12EN	RES1OVF11EN	RES1OVF10EN	RES1OVF9EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RES1OVF8EN	RES1OVF7EN	RES1OVF6EN	RES1OVF5EN	RES1OVF4EN	RES1OVF3EN	RES1OVF2EN	RES1OVF1EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-253. CHECKEVT2SEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RES1OVF16EN	R/W	0h	Enable CHECK16 RES1OVF as a source for CHECKEVT2. Reset type: SYSRSn
14	RES1OVF15EN	R/W	0h	Enable CHECK15 RES1OVF as a source for CHECKEVT2. Reset type: SYSRSn
13	RES1OVF14EN	R/W	0h	Enable CHECK14 RES1OVF as a source for CHECKEVT2. Reset type: SYSRSn
12	RES1OVF13EN	R/W	0h	Enable CHECK13 RES1OVF as a source for CHECKEVT2. Reset type: SYSRSn
11	RES1OVF12EN	R/W	0h	Enable CHECK12 RES1OVF as a source for CHECKEVT2. Reset type: SYSRSn
10	RES1OVF11EN	R/W	0h	Enable CHECK11 RES1OVF as a source for CHECKEVT2. Reset type: SYSRSn
9	RES1OVF10EN	R/W	0h	Enable CHECK10 RES1OVF as a source for CHECKEVT2. Reset type: SYSRSn
8	RES1OVF9EN	R/W	0h	Enable CHECK9 RES1OVF as a source for CHECKEVT2. Reset type: SYSRSn
7	RES1OVF8EN	R/W	0h	Enable CHECK8 RES1OVF as a source for CHECKEVT2. Reset type: SYSRSn
6	RES1OVF7EN	R/W	0h	Enable CHECK7 RES1OVF as a source for CHECKEVT2. Reset type: SYSRSn
5	RES1OVF6EN	R/W	0h	Enable CHECK6 RES1OVF as a source for CHECKEVT2. Reset type: SYSRSn
4	RES1OVF5EN	R/W	0h	Enable CHECK5 RES1OVF as a source for CHECKEVT2. Reset type: SYSRSn
3	RES1OVF4EN	R/W	0h	Enable CHECK4 RES1OVF as a source for CHECKEVT2. Reset type: SYSRSn

Table 24-253. CHECKEVT2SEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RES1OVF3EN	R/W	0h	Enable CHECK3 RES1OVF as a source for CHECKEVT2. Reset type: SYSRSn
1	RES1OVF2EN	R/W	0h	Enable CHECK2 RES1OVF as a source for CHECKEVT2. Reset type: SYSRSn
0	RES1OVF1EN	R/W	0h	Enable CHECK1 RES1OVF as a source for CHECKEVT2. Reset type: SYSRSn

24.16.5.16 CHECKEVT2SEL2 Register (Offset = 44h) [Reset = 00000000h]

CHECKEVT2SEL2 is shown in Figure 24-272 and described in Table 24-254.

Return to the Summary Table.

Checker X-Bar EVT2 Source Select Register 2

Figure 24-272. CHECKEVT2SEL2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RES2OVF16EN	RES2OVF15EN	RES2OVF14EN	RES2OVF13EN	RES2OVF12EN	RES2OVF11EN	RES2OVF10EN	RES2OVF9EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RES2OVF8EN	RES2OVF7EN	RES2OVF6EN	RES2OVF5EN	RES2OVF4EN	RES2OVF3EN	RES2OVF2EN	RES2OVF1EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-254. CHECKEVT2SEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RES2OVF16EN	R/W	0h	Enable CHECK16 RES2OVF as a source for CHECKEVT2. Reset type: SYSRSn
14	RES2OVF15EN	R/W	0h	Enable CHECK15 RES2OVF as a source for CHECKEVT2. Reset type: SYSRSn
13	RES2OVF14EN	R/W	0h	Enable CHECK14 RES2OVF as a source for CHECKEVT2. Reset type: SYSRSn
12	RES2OVF13EN	R/W	0h	Enable CHECK13 RES2OVF as a source for CHECKEVT2. Reset type: SYSRSn
11	RES2OVF12EN	R/W	0h	Enable CHECK12 RES2OVF as a source for CHECKEVT2. Reset type: SYSRSn
10	RES2OVF11EN	R/W	0h	Enable CHECK11 RES2OVF as a source for CHECKEVT2. Reset type: SYSRSn
9	RES2OVF10EN	R/W	0h	Enable CHECK10 RES2OVF as a source for CHECKEVT2. Reset type: SYSRSn
8	RES2OVF9EN	R/W	0h	Enable CHECK9 RES2OVF as a source for CHECKEVT2. Reset type: SYSRSn
7	RES2OVF8EN	R/W	0h	Enable CHECK8 RES2OVF as a source for CHECKEVT2. Reset type: SYSRSn
6	RES2OVF7EN	R/W	0h	Enable CHECK7 RES2OVF as a source for CHECKEVT2. Reset type: SYSRSn
5	RES2OVF6EN	R/W	0h	Enable CHECK6 RES2OVF as a source for CHECKEVT2. Reset type: SYSRSn
4	RES2OVF5EN	R/W	0h	Enable CHECK5 RES2OVF as a source for CHECKEVT2. Reset type: SYSRSn
3	RES2OVF4EN	R/W	0h	Enable CHECK4 RES2OVF as a source for CHECKEVT2. Reset type: SYSRSn

Table 24-254. CHECKEVT2SEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RES2OVF3EN	R/W	0h	Enable CHECK3 RES2OVF as a source for CHECKEVT2. Reset type: SYSRSn
1	RES2OVF2EN	R/W	0h	Enable CHECK2 RES2OVF as a source for CHECKEVT2. Reset type: SYSRSn
0	RES2OVF1EN	R/W	0h	Enable CHECK1 RES2OVF as a source for CHECKEVT2. Reset type: SYSRSn

24.16.5.17 CHECKEVT2SEL3 Register (Offset = 48h) [Reset = 00000000h]

CHECKEVT2SEL3 is shown in Figure 24-273 and described in Table 24-255.

Return to the Summary Table.

Checker X-Bar EVT2 Source Select Register 3

Figure 24-273. CHECKEVT2SEL3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
OOT16EN	OOT15EN	OOT14EN	OOT13EN	OOT12EN	OOT11EN	OOT10EN	OOT9EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
OOT8EN	OOT7EN	OOT6EN	OOT5EN	OOT4EN	OOT3EN	OOT2EN	OOT1EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-255. CHECKEVT2SEL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	OOT16EN	R/W	0h	Enable CHECK16 OOT as a source for CHECKEVT2. Reset type: SYSRSn
14	OOT15EN	R/W	0h	Enable CHECK15 OOTas a source for CHECKEVT2. Reset type: SYSRSn
13	OOT14EN	R/W	0h	Enable CHECK14 OOT as a source for CHECKEVT2. Reset type: SYSRSn
12	OOT13EN	R/W	0h	Enable CHECK13 OOT as a source for CHECKEVT2. Reset type: SYSRSn
11	OOT12EN	R/W	0h	Enable CHECK12 OOT as a source for CHECKEVT2. Reset type: SYSRSn
10	OOT11EN	R/W	0h	Enable CHECK11 OOTas a source for CHECKEVT2. Reset type: SYSRSn
9	OOT10EN	R/W	0h	Enable CHECK10 OOT as a source for CHECKEVT2. Reset type: SYSRSn
8	OOT9EN	R/W	0h	Enable CHECK9 OOT as a source for CHECKEVT2. Reset type: SYSRSn
7	OOT8EN	R/W	0h	Enable CHECK8 OOT as a source for CHECKEVT2. Reset type: SYSRSn
6	OOT7EN	R/W	0h	Enable CHECK7 OOTas a source for CHECKEVT2. Reset type: SYSRSn
5	OOT6EN	R/W	0h	Enable CHECK6 OOT as a source for CHECKEVT2. Reset type: SYSRSn
4	OOT5EN	R/W	0h	Enable CHECK5 OOT as a source for CHECKEVT2. Reset type: SYSRSn
3	OOT4EN	R/W	0h	Enable CHECK4 OOT as a source for CHECKEVT2. Reset type: SYSRSn

Table 24-255. CHECKEVT2SEL3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	OOT3EN	R/W	0h	Enable CHECK3 OOTas a source for CHECKEVT2. Reset type: SYSRSn
1	OOT2EN	R/W	0h	Enable CHECK2 OOT as a source for CHECKEVT2. Reset type: SYSRSn
0	OOT1EN	R/W	0h	Enable CHECK1 OOT as a source for CHECKEVT2. Reset type: SYSRSn

24.16.5.18 CHECKEVT3SEL1 Register (Offset = 50h) [Reset = 00000000h]

CHECKEVT3SEL1 is shown in Figure 24-274 and described in Table 24-256.

Return to the Summary Table.

Checker X-Bar EVT3 Source Select Register 1

Figure 24-274. CHECKEVT3SEL1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RES1OVF16EN	RES1OVF15EN	RES1OVF14EN	RES1OVF13EN	RES1OVF12EN	RES1OVF11EN	RES1OVF10EN	RES1OVF9EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RES1OVF8EN	RES1OVF7EN	RES1OVF6EN	RES1OVF5EN	RES1OVF4EN	RES1OVF3EN	RES1OVF2EN	RES1OVF1EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-256. CHECKEVT3SEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RES1OVF16EN	R/W	0h	Enable CHECK16 RES1OVF as a source for CHECKEVT3. Reset type: SYSRSn
14	RES1OVF15EN	R/W	0h	Enable CHECK15 RES1OVF as a source for CHECKEVT3. Reset type: SYSRSn
13	RES1OVF14EN	R/W	0h	Enable CHECK14 RES1OVF as a source for CHECKEVT3. Reset type: SYSRSn
12	RES1OVF13EN	R/W	0h	Enable CHECK13 RES1OVF as a source for CHECKEVT3. Reset type: SYSRSn
11	RES1OVF12EN	R/W	0h	Enable CHECK12 RES1OVF as a source for CHECKEVT3. Reset type: SYSRSn
10	RES1OVF11EN	R/W	0h	Enable CHECK11 RES1OVF as a source for CHECKEVT3. Reset type: SYSRSn
9	RES1OVF10EN	R/W	0h	Enable CHECK10 RES1OVF as a source for CHECKEVT3. Reset type: SYSRSn
8	RES1OVF9EN	R/W	0h	Enable CHECK9 RES1OVF as a source for CHECKEVT3. Reset type: SYSRSn
7	RES1OVF8EN	R/W	0h	Enable CHECK8 RES1OVF as a source for CHECKEVT3. Reset type: SYSRSn
6	RES1OVF7EN	R/W	0h	Enable CHECK7 RES1OVF as a source for CHECKEVT3. Reset type: SYSRSn
5	RES1OVF6EN	R/W	0h	Enable CHECK6 RES1OVF as a source for CHECKEVT3. Reset type: SYSRSn
4	RES1OVF5EN	R/W	0h	Enable CHECK5 RES1OVF as a source for CHECKEVT3. Reset type: SYSRSn
3	RES1OVF4EN	R/W	0h	Enable CHECK4 RES1OVF as a source for CHECKEVT3. Reset type: SYSRSn

Table 24-256. CHECKEVT3SEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RES1OVF3EN	R/W	0h	Enable CHECK3 RES1OVF as a source for CHECKEVT3. Reset type: SYSRSn
1	RES1OVF2EN	R/W	0h	Enable CHECK2 RES1OVF as a source for CHECKEVT3. Reset type: SYSRSn
0	RES1OVF1EN	R/W	0h	Enable CHECK1 RES1OVF as a source for CHECKEVT3. Reset type: SYSRSn

24.16.5.19 CHECKEVT3SEL2 Register (Offset = 54h) [Reset = 00000000h]

CHECKEVT3SEL2 is shown in Figure 24-275 and described in Table 24-257.

Return to the Summary Table.

Checker X-Bar EVT3 Source Select Register 2

Figure 24-275. CHECKEVT3SEL2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RES2OVF16EN	RES2OVF15EN	RES2OVF14EN	RES2OVF13EN	RES2OVF12EN	RES2OVF11EN	RES2OVF10EN	RES2OVF9EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RES2OVF8EN	RES2OVF7EN	RES2OVF6EN	RES2OVF5EN	RES2OVF4EN	RES2OVF3EN	RES2OVF2EN	RES2OVF1EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-257. CHECKEVT3SEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RES2OVF16EN	R/W	0h	Enable CHECK16 RES2OVF as a source for CHECKEVT3. Reset type: SYSRSn
14	RES2OVF15EN	R/W	0h	Enable CHECK15 RES2OVF as a source for CHECKEVT3. Reset type: SYSRSn
13	RES2OVF14EN	R/W	0h	Enable CHECK14 RES2OVF as a source for CHECKEVT3. Reset type: SYSRSn
12	RES2OVF13EN	R/W	0h	Enable CHECK13 RES2OVF as a source for CHECKEVT3. Reset type: SYSRSn
11	RES2OVF12EN	R/W	0h	Enable CHECK12 RES2OVF as a source for CHECKEVT3. Reset type: SYSRSn
10	RES2OVF11EN	R/W	0h	Enable CHECK11 RES2OVF as a source for CHECKEVT3. Reset type: SYSRSn
9	RES2OVF10EN	R/W	0h	Enable CHECK10 RES2OVF as a source for CHECKEVT3. Reset type: SYSRSn
8	RES2OVF9EN	R/W	0h	Enable CHECK9 RES2OVF as a source for CHECKEVT3. Reset type: SYSRSn
7	RES2OVF8EN	R/W	0h	Enable CHECK8 RES2OVF as a source for CHECKEVT3. Reset type: SYSRSn
6	RES2OVF7EN	R/W	0h	Enable CHECK7 RES2OVF as a source for CHECKEVT3. Reset type: SYSRSn
5	RES2OVF6EN	R/W	0h	Enable CHECK6 RES2OVF as a source for CHECKEVT3. Reset type: SYSRSn
4	RES2OVF5EN	R/W	0h	Enable CHECK5 RES2OVF as a source for CHECKEVT3. Reset type: SYSRSn
3	RES2OVF4EN	R/W	0h	Enable CHECK4 RES2OVF as a source for CHECKEVT3. Reset type: SYSRSn

Table 24-257. CHECKEVT3SEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RES2OVF3EN	R/W	0h	Enable CHECK3 RES2OVF as a source for CHECKEVT3. Reset type: SYSRSn
1	RES2OVF2EN	R/W	0h	Enable CHECK2 RES2OVF as a source for CHECKEVT3. Reset type: SYSRSn
0	RES2OVF1EN	R/W	0h	Enable CHECK1 RES2OVF as a source for CHECKEVT3. Reset type: SYSRSn

24.16.5.20 CHECKEVT3SEL3 Register (Offset = 58h) [Reset = 00000000h]

CHECKEVT3SEL3 is shown in Figure 24-276 and described in Table 24-258.

Return to the Summary Table.

Checker X-Bar EVT3 Source Select Register 3

Figure 24-276. CHECKEVT3SEL3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
OOT16EN	OOT15EN	OOT14EN	OOT13EN	OOT12EN	OOT11EN	OOT10EN	OOT9EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
OOT8EN	OOT7EN	OOT6EN	OOT5EN	OOT4EN	OOT3EN	OOT2EN	OOT1EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-258. CHECKEVT3SEL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	OOT16EN	R/W	0h	Enable CHECK16 OOT as a source for CHECKEVT3. Reset type: SYSRSn
14	OOT15EN	R/W	0h	Enable CHECK15 OOTas a source for CHECKEVT3. Reset type: SYSRSn
13	OOT14EN	R/W	0h	Enable CHECK14 OOT as a source for CHECKEVT3. Reset type: SYSRSn
12	OOT13EN	R/W	0h	Enable CHECK13 OOT as a source for CHECKEVT3. Reset type: SYSRSn
11	OOT12EN	R/W	0h	Enable CHECK12 OOT as a source for CHECKEVT3. Reset type: SYSRSn
10	OOT11EN	R/W	0h	Enable CHECK11 OOTas a source for CHECKEVT3. Reset type: SYSRSn
9	OOT10EN	R/W	0h	Enable CHECK10 OOT as a source for CHECKEVT3. Reset type: SYSRSn
8	OOT9EN	R/W	0h	Enable CHECK9 OOT as a source for CHECKEVT3. Reset type: SYSRSn
7	OOT8EN	R/W	0h	Enable CHECK8 OOT as a source for CHECKEVT3. Reset type: SYSRSn
6	OOT7EN	R/W	0h	Enable CHECK7 OOTas a source for CHECKEVT3. Reset type: SYSRSn
5	OOT6EN	R/W	0h	Enable CHECK6 OOT as a source for CHECKEVT3. Reset type: SYSRSn
4	OOT5EN	R/W	0h	Enable CHECK5 OOT as a source for CHECKEVT3. Reset type: SYSRSn
3	OOT4EN	R/W	0h	Enable CHECK4 OOT as a source for CHECKEVT3. Reset type: SYSRSn

Table 24-258. CHECKEVT3SEL3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	OOT3EN	R/W	0h	Enable CHECK3 OOTas a source for CHECKEVT3. Reset type: SYSRSn
1	OOT2EN	R/W	0h	Enable CHECK2 OOT as a source for CHECKEVT3. Reset type: SYSRSn
0	OOT1EN	R/W	0h	Enable CHECK1 OOT as a source for CHECKEVT3. Reset type: SYSRSn

24.16.5.21 CHECKEVT4SEL1 Register (Offset = 60h) [Reset = 00000000h]

CHECKEVT4SEL1 is shown in Figure 24-277 and described in Table 24-259.

Return to the Summary Table.

Checker X-Bar EVT4 Source Select Register 1

Figure 24-277. CHECKEVT4SEL1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RES1OVF16EN	RES1OVF15EN	RES1OVF14EN	RES1OVF13EN	RES1OVF12EN	RES1OVF11EN	RES1OVF10EN	RES1OVF9EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RES1OVF8EN	RES1OVF7EN	RES1OVF6EN	RES1OVF5EN	RES1OVF4EN	RES1OVF3EN	RES1OVF2EN	RES1OVF1EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-259. CHECKEVT4SEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RES1OVF16EN	R/W	0h	Enable CHECK16 RES1OVF as a source for CHECKEVT4. Reset type: SYSRSn
14	RES1OVF15EN	R/W	0h	Enable CHECK15 RES1OVF as a source for CHECKEVT4. Reset type: SYSRSn
13	RES1OVF14EN	R/W	0h	Enable CHECK14 RES1OVF as a source for CHECKEVT4. Reset type: SYSRSn
12	RES1OVF13EN	R/W	0h	Enable CHECK13 RES1OVF as a source for CHECKEVT4. Reset type: SYSRSn
11	RES1OVF12EN	R/W	0h	Enable CHECK12 RES1OVF as a source for CHECKEVT4. Reset type: SYSRSn
10	RES1OVF11EN	R/W	0h	Enable CHECK11 RES1OVF as a source for CHECKEVT4. Reset type: SYSRSn
9	RES1OVF10EN	R/W	0h	Enable CHECK10 RES1OVF as a source for CHECKEVT4. Reset type: SYSRSn
8	RES1OVF9EN	R/W	0h	Enable CHECK9 RES1OVF as a source for CHECKEVT4. Reset type: SYSRSn
7	RES1OVF8EN	R/W	0h	Enable CHECK8 RES1OVF as a source for CHECKEVT4. Reset type: SYSRSn
6	RES1OVF7EN	R/W	0h	Enable CHECK7 RES1OVF as a source for CHECKEVT4. Reset type: SYSRSn
5	RES1OVF6EN	R/W	0h	Enable CHECK6 RES1OVF as a source for CHECKEVT4. Reset type: SYSRSn
4	RES1OVF5EN	R/W	0h	Enable CHECK5 RES1OVF as a source for CHECKEVT4. Reset type: SYSRSn
3	RES1OVF4EN	R/W	0h	Enable CHECK4 RES1OVF as a source for CHECKEVT4. Reset type: SYSRSn

Table 24-259. CHECKEVT4SEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RES1OVF3EN	R/W	0h	Enable CHECK3 RES1OVF as a source for CHECKEVT4. Reset type: SYSRSn
1	RES1OVF2EN	R/W	0h	Enable CHECK2 RES1OVF as a source for CHECKEVT4. Reset type: SYSRSn
0	RES1OVF1EN	R/W	0h	Enable CHECK1 RES1OVF as a source for CHECKEVT4. Reset type: SYSRSn

24.16.5.22 CHECKEVT4SEL2 Register (Offset = 64h) [Reset = 00000000h]

CHECKEVT4SEL2 is shown in Figure 24-278 and described in Table 24-260.

Return to the Summary Table.

Checker X-Bar EVT4 Source Select Register 2

Figure 24-278. CHECKEVT4SEL2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RES2OVF16EN	RES2OVF15EN	RES2OVF14EN	RES2OVF13EN	RES2OVF12EN	RES2OVF11EN	RES2OVF10EN	RES2OVF9EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RES2OVF8EN	RES2OVF7EN	RES2OVF6EN	RES2OVF5EN	RES2OVF4EN	RES2OVF3EN	RES2OVF2EN	RES2OVF1EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-260. CHECKEVT4SEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RES2OVF16EN	R/W	0h	Enable CHECK16 RES2OVF as a source for CHECKEVT4. Reset type: SYSRSn
14	RES2OVF15EN	R/W	0h	Enable CHECK15 RES2OVF as a source for CHECKEVT4. Reset type: SYSRSn
13	RES2OVF14EN	R/W	0h	Enable CHECK14 RES2OVF as a source for CHECKEVT4. Reset type: SYSRSn
12	RES2OVF13EN	R/W	0h	Enable CHECK13 RES2OVF as a source for CHECKEVT4. Reset type: SYSRSn
11	RES2OVF12EN	R/W	0h	Enable CHECK12 RES2OVF as a source for CHECKEVT4. Reset type: SYSRSn
10	RES2OVF11EN	R/W	0h	Enable CHECK11 RES2OVF as a source for CHECKEVT4. Reset type: SYSRSn
9	RES2OVF10EN	R/W	0h	Enable CHECK10 RES2OVF as a source for CHECKEVT4. Reset type: SYSRSn
8	RES2OVF9EN	R/W	0h	Enable CHECK9 RES2OVF as a source for CHECKEVT4. Reset type: SYSRSn
7	RES2OVF8EN	R/W	0h	Enable CHECK8 RES2OVF as a source for CHECKEVT4. Reset type: SYSRSn
6	RES2OVF7EN	R/W	0h	Enable CHECK7 RES2OVF as a source for CHECKEVT4. Reset type: SYSRSn
5	RES2OVF6EN	R/W	0h	Enable CHECK6 RES2OVF as a source for CHECKEVT4. Reset type: SYSRSn
4	RES2OVF5EN	R/W	0h	Enable CHECK5 RES2OVF as a source for CHECKEVT4. Reset type: SYSRSn
3	RES2OVF4EN	R/W	0h	Enable CHECK4 RES2OVF as a source for CHECKEVT4. Reset type: SYSRSn

Table 24-260. CHECKEVT4SEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RES2OVF3EN	R/W	0h	Enable CHECK3 RES2OVF as a source for CHECKEVT4. Reset type: SYSRSn
1	RES2OVF2EN	R/W	0h	Enable CHECK2 RES2OVF as a source for CHECKEVT4. Reset type: SYSRSn
0	RES2OVF1EN	R/W	0h	Enable CHECK1 RES2OVF as a source for CHECKEVT4. Reset type: SYSRSn

24.16.5.23 CHECKEVT4SEL3 Register (Offset = 68h) [Reset = 00000000h]

CHECKEVT4SEL3 is shown in Figure 24-279 and described in Table 24-261.

Return to the Summary Table.

Checker X-Bar EVT4 Source Select Register 3

Figure 24-279. CHECKEVT4SEL3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
OOT16EN	OOT15EN	OOT14EN	OOT13EN	OOT12EN	OOT11EN	OOT10EN	OOT9EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
OOT8EN	OOT7EN	OOT6EN	OOT5EN	OOT4EN	OOT3EN	OOT2EN	OOT1EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-261. CHECKEVT4SEL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	OOT16EN	R/W	0h	Enable CHECK16 OOT as a source for CHECKEVT4. Reset type: SYSRSn
14	OOT15EN	R/W	0h	Enable CHECK15 OOTas a source for CHECKEVT4. Reset type: SYSRSn
13	OOT14EN	R/W	0h	Enable CHECK14 OOT as a source for CHECKEVT4. Reset type: SYSRSn
12	OOT13EN	R/W	0h	Enable CHECK13 OOT as a source for CHECKEVT4. Reset type: SYSRSn
11	OOT12EN	R/W	0h	Enable CHECK12 OOT as a source for CHECKEVT4. Reset type: SYSRSn
10	OOT11EN	R/W	0h	Enable CHECK11 OOTas a source for CHECKEVT4. Reset type: SYSRSn
9	OOT10EN	R/W	0h	Enable CHECK10 OOT as a source for CHECKEVT4. Reset type: SYSRSn
8	OOT9EN	R/W	0h	Enable CHECK9 OOT as a source for CHECKEVT4. Reset type: SYSRSn
7	OOT8EN	R/W	0h	Enable CHECK8 OOT as a source for CHECKEVT4. Reset type: SYSRSn
6	OOT7EN	R/W	0h	Enable CHECK7 OOTas a source for CHECKEVT4. Reset type: SYSRSn
5	OOT6EN	R/W	0h	Enable CHECK6 OOT as a source for CHECKEVT4. Reset type: SYSRSn
4	OOT5EN	R/W	0h	Enable CHECK5 OOT as a source for CHECKEVT4. Reset type: SYSRSn
3	OOT4EN	R/W	0h	Enable CHECK4 OOT as a source for CHECKEVT4. Reset type: SYSRSn

Table 24-261. CHECKEVT4SEL3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	OOT3EN	R/W	0h	Enable CHECK3 OOTas a source for CHECKEVT4. Reset type: SYSRSn
1	OOT2EN	R/W	0h	Enable CHECK2 OOT as a source for CHECKEVT4. Reset type: SYSRSn
0	OOT1EN	R/W	0h	Enable CHECK1 OOT as a source for CHECKEVT4. Reset type: SYSRSn

24.16.6 ADC_GLOBAL_REGS Registers

Table 24-262 lists the memory-mapped registers for the ADC_GLOBAL_REGS registers. All register offset addresses not listed in Table 24-262 should be considered as reserved locations and the register contents should not be modified.

Table 24-262. ADC_GLOBAL_REGS Registers

Offset	Acronym	Register Name	Protection
0h	ADCSOCFRCGB	ADC Global SOC Force	PARITY
4h	ADCSOCFRCGBSEL	ADC Global SOC Force Select	PARITY
8h	PARITY_TEST_ALT1	Enables parity test	

Complex bit access types are encoded to fit into small table cells. Table 24-263 shows the codes that are used for access types in this section.

Table 24-263. ADC_GLOBAL_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

24.16.6.1 ADCSOCFRCGB Register (Offset = 0h) [Reset = 00000000h]

ADCSOCFRCGB is shown in Figure 24-280 and described in Table 24-264.

Return to the Summary Table.

ADC Global SOC Force

Figure 24-280. ADCSOCFRCGB Register

31	30	29	28	27	26	25	24
SOC31	SOC30	SOC29	SOC28	SOC27	SOC26	SOC25	SOC24
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
23	22	21	20	19	18	17	16
SOC23	SOC22	SOC21	SOC20	SOC19	SOC18	SOC17	SOC16
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
SOC7	SOC6	SOC5	SOC4	SOC3	SOC2	SOC1	SOC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 24-264. ADCSOCFRCGB Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SOC31	R-0/W1S	0h	Indicate if SOC31 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
30	SOC30	R-0/W1S	0h	Indicate if SOC30 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
29	SOC29	R-0/W1S	0h	Indicate if SOC29 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
28	SOC28	R-0/W1S	0h	Indicate if SOC28 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
27	SOC27	R-0/W1S	0h	Indicate if SOC27 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
26	SOC26	R-0/W1S	0h	Indicate if SOC26 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
25	SOC25	R-0/W1S	0h	Indicate if SOC25 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn

Table 24-264. ADCSOCFRCGB Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	SOC24	R-0/W1S	0h	Indicate if SOC24 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
23	SOC23	R-0/W1S	0h	Indicate if SOC23 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
22	SOC22	R-0/W1S	0h	Indicate if SOC22 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
21	SOC21	R-0/W1S	0h	Indicate if SOC21 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
20	SOC20	R-0/W1S	0h	Indicate if SOC20 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
19	SOC19	R-0/W1S	0h	Indicate if SOC19 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
18	SOC18	R-0/W1S	0h	Indicate if SOC18 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
17	SOC17	R-0/W1S	0h	Indicate if SOC17 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
16	SOC16	R-0/W1S	0h	Indicate if SOC16 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
15	SOC15	R-0/W1S	0h	Indicate if SOC15 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
14	SOC14	R-0/W1S	0h	Indicate if SOC14 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
13	SOC13	R-0/W1S	0h	Indicate if SOC13 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
12	SOC12	R-0/W1S	0h	Indicate if SOC12 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
11	SOC11	R-0/W1S	0h	Indicate if SOC11 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn

Table 24-264. ADCSOCFRCGB Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SOC10	R-0/W1S	0h	Indicate if SOC10 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
9	SOC9	R-0/W1S	0h	Indicate if SOC9 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
8	SOC8	R-0/W1S	0h	Indicate if SOC8 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
7	SOC7	R-0/W1S	0h	Indicate if SOC7 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
6	SOC6	R-0/W1S	0h	Indicate if SOC6 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
5	SOC5	R-0/W1S	0h	Indicate if SOC5 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
4	SOC4	R-0/W1S	0h	Indicate if SOC4 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
3	SOC3	R-0/W1S	0h	Indicate if SOC3 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
2	SOC2	R-0/W1S	0h	Indicate if SOC2 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
1	SOC1	R-0/W1S	0h	Indicate if SOC1 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn
0	SOC0	R-0/W1S	0h	Indicate if SOC0 selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: SYSRSn

24.16.6.2 ADCSOCFRCGBSEL Register (Offset = 4h) [Reset = 00000000h]

ADCSOCFRCGBSEL is shown in Figure 24-281 and described in Table 24-265.

Return to the Summary Table.

ADC Global SOC Force Select

Figure 24-281. ADCSOCFRCGBSEL Register

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

Table 24-265. ADCSOCFRCGBSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R-0	0h	Reserved
4	ADCE	R/W	0h	Indicate if ADCE selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: XRSn
3	ADCD	R/W	0h	Indicate if ADCD selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: XRSn
2	ADCC	R/W	0h	Indicate if ADCC selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: XRSn
1	ADCB	R/W	0h	Indicate if ADCB selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: XRSn
0	ADCA	R/W	0h	Indicate if ADCA selected for global SW trigger 0 : Not selected for Global SW Trigger 1 : Selected for Global SW Trigger Reset type: XRSn

24.16.6.3 PARITY_TEST_ALT1 Register (Offset = 8h) [Reset = 00000000h]

PARITY_TEST_ALT1 is shown in Figure 24-282 and described in Table 24-266.

Return to the Summary Table.

Enables parity test

Figure 24-282. PARITY_TEST_ALT1 Register

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

Table 24-266. PARITY_TEST_ALT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	TESTEN	R/W	0h	1010: Parity test feature is enabled All other values: Parity test feature is disabled Note: (1) When the parity test feature is enabled, actual registers are not accessible in the memory map. Instead, the parity values (final XOR output indicating the parity error) are accessible. Parity is computed for every byte and the corresponding parity error value is available at the bit-0 of every byte. Value of '1' written to the parity bit after enabling the parity test feature can be used to inject the error by inverting the stored parity value. (2) It is recommended to leave the field as 0101 or 0000 after completing the parity test. Reset type: SYSRSn

Chapter 25

Buffered Digital-to-Analog Converter (DAC)

The buffered digital-to-analog converter (DAC) is an analog module that can output a programmable, arbitrary reference voltage.

25.1 Introduction

The buffered DAC module consists of an internal 12-bit DAC and an analog output buffer that is capable of driving an external load. For driving even higher loads than typical, a trade-off can be made between load size and output voltage swing. For the load conditions of the buffered DAC, see the device-specific data sheet. The buffered DAC is a general-purpose DAC that can be used to generate a DC voltage in addition to AC waveforms such as sine waves, square waves, triangle waves and so forth. Software writes to the DAC value register can take effect immediately or can be synchronized with EPWMSYNCPER events.

25.1.1 DAC Related Collateral

Foundational Materials

- C28x Academy - DAC
- C29x Academy - Digital to Analog Converter (DAC)
- *High Speed, Digital to Analog Converters Basics Application Report*
- *Real-Time Control Reference Guide*
  - Refer to the DAC section
- *Understanding Data Converters Application Report*

Getting Started Materials

- MathWorks F2807x/F2837xD/F2837xS/F28004x/F2838x DAC
  - NOTE: This is a non-TI (third party) site.

25.1.2 Features

Each buffered DAC has the following features:

- 12-bit programmable internal DAC
- Selectable reference voltage source
- x1 and x2 gain modes when using internal VREFHI
- Ability to synchronize with EPWMSYNCPER

25.1.3 Block Diagram

The block diagram for the buffered DAC is shown in Figure 25-1.

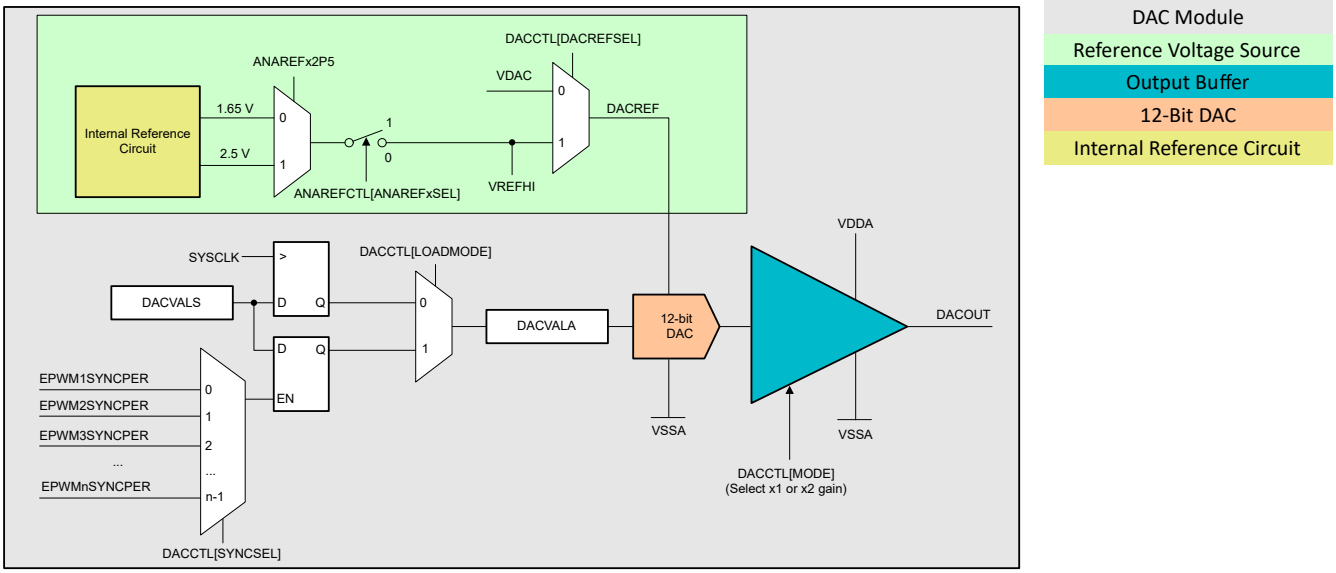


Figure 25-1. DAC Module Block Diagram

25.2 Using the DAC

The internal DAC's reference voltage source, DACREF, is selectable between VDAC and VREFHI. The x2 gain mode is only available when VREFHI is set as DACREF and internal reference mode is used, which can be configured by DACCTL[MODE] register. The internal reference circuit generates 1.65V and 2.5V. To select either 2.5V or 1.65V, configure ANAREF2P5 register and set ANAREFSEL register to 0 (see the *Analog Subsystem* chapter on how to switch to internal reference mode). Even though the buffered DAC has an x2 gain mode, the maximum output voltage from the buffered DAC is not greater than VDDA. Table 25-1 lists the gain mode combinations supported by the buffered DAC. In this table, x = A or B, X = Don't Care, VDAC/ VREFHI = 2.5v, VDDA = 3.3v, and DACVAL = 4095.

Table 25-1. DAC Supported Gain Mode Combinations

DACREFSEL	ANAREFSEL	ANAREF2P5	Reference Source	Reference Voltage (V)	Mode	Maximum DAC Output (V)	Support Status
0	X	X	External	VDAC	0	2.5	Supported
0	X	X	External	VDAC	1	2.5	Not Supported
1	0	0	Internal	1.65	0	1.65	Not Supported
1	0	0	Internal	1.65	1	3.3	Supported
1	0	1	Internal	2.5	0	2.5	Supported
1	0	1	Internal	2.5	1	2.5	Not Supported
1	1	X	External	VREFHI	0	2.5	Supported
1	1	X	External	VREFHI	1	2.5	Not Supported

Two sets of DACVAL registers, DACVALA and DACVALS, are present in the buffered DAC module. DACVALA is a read-only register that actively controls the buffered DAC value. DACVALS is a writable shadow register that loads into DACVALA either immediately or synchronized with the next EPWMSYNCPER event. If the clock to the buffered DAC is disabled while the buffered DAC is outputting a voltage, the output voltage remains unaffected, but DACVALA and DACVALS is no longer updated with register writes. Enabling the clock to the buffered DAC restores the DAC to the state before the clock was disabled.

The output of the internal DAC is calculated with the following equation:

$$DACOUT = \frac{DACVALA * DACREF}{4096} \quad (20)$$

The output buffer of the buffered DAC can exhibit non-linear behavior near the supply rails (VDDA/VSSA). To determine the linear range of the buffered DAC, see the device-specific data sheet.

25.2.1 Initialization Sequence

1. Enable the buffered DAC clock.
2. Set DACREF with DACREFSEL.
3. Power up the buffered DAC with DACOUTEN.
4. Wait for the power-up time to elapse before outputting a voltage. To determine the power-up time of the buffered DAC, see the device data sheet.
5. For predictable behavior of the buffered DAC, consecutive writes to DACVALS must be spaced apart according to the settling time of the buffered DAC. To determine the settling time of the buffered DAC, see the device data sheet.

25.2.2 DAC Offset Adjustment

Zero offset error is defined as the difference between the voltage at midcode (2048) and 1.25V (for 2.5V reference voltage). DAC offset error is calibrated at 2.5V reference voltage and loaded into the DAC offset trim register as part of the Device_cal() function. If the DAC is used at any reference voltage other than 2.5V, the offset trim must be adjusted to make sure the offset error performance stays within the device-specific data sheet limits. The DAC offset register is a 16-bit register that contains the 8-bit signed offset trim in the lower half of the register. Use the function call DAC_tuneOffsetTrim() found in C2000Ware to adjust the offset.

25.2.3 EPWMSYNCPER Signal

EPWMSYNCPER comes from the Time-Base submodule of the EPWM. For a detailed description of how this signal is generated, refer to the *Time-Base Submodule* section of the *Enhanced Pulse Width Modulator (ePWM)* chapter.

The EPWMSYNCPER signal that loads DACVALA when DACCTL [LOADMODE] = 1 is a level trigger load. If TBCTR and TBPRD of the EPWM are both 0, EPWMSYNCPER is held at a high level and DACVALA is immediately loaded from DACVALS irrespective of the value of DACCTL [LOADMODE]. Due to this, configure the EPWM first before setting DACCTL [LOADMODE] to 1.

Note

The name of the sync signal that the GPDAC receives from the EPWM has been updated from PWMSYNC to EPWMSYNCPER (SYNCPER/PWMSYNCPER/EPWMxSYNCPER) to avoid confusion with the other EPWM sync signals EPWMSYNCI and EPWMSYNCO. For a description of these signals, see the *Enhanced Pulse Width Modulator (ePWM)* chapter.

25.3 Lock Registers

A DACLOCK register is provided to prevent spurious writes from modifying the DACCTL, DACVALS, and DACOUTEN registers. Once a register is protected through DACLOCK, write access are locked out until the device is reset.

25.4 Software

25.4.1 DAC Registers to Driverlib Functions

Table 25-2. DAC Registers to Driverlib Functions

File	Driverlib Function
DACREV	
dac.h	DAC_getRevision
DACCTL	
dac.h	DAC_setReferenceVoltage
dac.h	DAC_setGainMode
dac.h	DAC_setLoadMode
dac.h	DAC_setPWMSyncSignal
DACVALA	
dac.h	DAC_getActiveValue
DACVALS	
dac.h	DAC_setShadowValue
dac.h	DAC_getShadowValue
DACOUTEN	
dac.h	DAC_enableOutput
dac.h	DAC_disableOutput
DACLOCK	
dac.h	DAC_lockRegister
dac.h	DAC_isRegisterLocked
DACTRIM	
dac.c	DAC_tuneOffsetTrim
dac.h	DAC_setOffsetTrim
dac.h	DAC_getOffsetTrim

25.4.2 DAC Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/dac

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

25.4.2.1 Buffered DAC Enable - SINGLE_CORE

FILE: buffdac_ex1_enable.c

This example generates a voltage on the buffered DAC output, DACOUTA/ADCINA0 and uses the default DAC reference setting of VDAC.

External Connections

- When the DAC reference is set to VDAC, an external reference voltage must be applied to the VDAC pin. This can be accomplished by connecting a jumper wire from 3.3V to ADCINB0.

Watch Variables

- None.

25.4.2.2 Buffered DAC Random - SINGLE_CORE

FILE: buffdac_ex2_random.c

This example generates random voltages on the buffered DAC output, DACOUTA/ADCINA0 and uses the default DAC reference setting of VDAC.

External Connections

- When the DAC reference is set to VDAC, an external reference voltage must be applied to the VDAC pin. This can accomplished by connecting a jumper wire from 3.3V to ADCINB0.

Watch Variables

- None.

25.5 DAC Registers

This Section describes the DAC Registers.

25.5.1 DAC Base Address Table

Table 25-3. DAC Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
DAC_REGS	DACA_BASE	0x700D_0000	YES	YES	YES	YES	YES	YES	-	YES
DAC_REGS	DACB_BASE	0x700D_1000	YES	YES	YES	YES	YES	YES	-	YES

25.5.2 DAC_REGS Registers

Table 25-4 lists the memory-mapped registers for the DAC_REGS registers. All register offset addresses not listed in Table 25-4 should be considered as reserved locations and the register contents should not be modified.

Table 25-4. DAC_REGS Registers

Offset	Acronym	Register Name	Protection
0h	DACREV	DAC Revision Register	
2h	DACCTL	DAC Control Register	
4h	DACVALA	DAC Value Register - Active	
6h	DACVALS	DAC Value Register - Shadow	
8h	DACOUTEN	DAC Output Enable Register	
Ah	DACLOCK	DAC Lock Register	
Ch	DACTRIM	DAC Trim Register	

Complex bit access types are encoded to fit into small table cells. Table 25-5 shows the codes that are used for access types in this section.

Table 25-5. DAC_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value

25.5.2.1 DACREV Register (Offset = 0h) [Reset = 0000h]

DACREV is shown in Figure 25-2 and described in Table 25-6.

Return to the Summary Table.

DAC Revision Register

Figure 25-2. DACREV Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
REV							
R-0h							

Table 25-6. DACREV Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	REV	R	0h	DAC Revision Reset type: SYSRSn

25.5.2.2 DACCTL Register (Offset = 2h) [Reset = 0000h]

DACCTL is shown in Figure 25-3 and described in Table 25-7.

Return to the Summary Table.

DAC Control Register

Figure 25-3. DACCTL Register

15	14	13	12	11	10	9	8
RESERVED							SYNCSEL
R-0h							R/W-0h
7	6	5	4	3	2	1	0
SYNCSEL				RESERVED	LOADMODE	MODE	DACREFSEL
R/W-0h				R-0h	R/W-0h	R/W-0h	R/W-0h

Table 25-7. DACCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8-4	SYNCSEL	R/W	0h	DAC EPWMSYNCPER select. Determines which EPWMSYNCPER signal will update the DACVALA register. Where n represents the maximum number of EPWMSYNCPER signals available on the device: 0 EPWM1SYNCPER 1 EPWM2SYNCPER 2 EPWM3SYNCPER ... n-1 EPWMnSYNCPER Reset type: SYSRSn
3	RESERVED	R	0h	Reserved
2	LOADMODE	R/W	0h	DACVALA load mode. Determines when the DACVALA register is updated with the value from DACVALS. 0 Load on next SYSCLK 1 Load on next EPWMSYNCPER specified by SYNCSEL Reset type: SYSRSn
1	MODE	R/W	0h	DAC gain mode select. Selects the gain mode for the buffered output. The MODE value is only used when DACREFSEL=1 and internal ADC reference mode is selected. 0 Gain is 1 1 Gain is 2 Reset type: SYSRSn
0	DACREFSEL	R/W	0h	DAC reference select. Selects which voltage references are used by the DAC. 0 VDAC/VSSA are the reference voltages 1 ADC VREFHI/VSSA are the reference voltages Reset type: SYSRSn

25.5.2.3 DACVALA Register (Offset = 4h) [Reset = 0000h]

DACVALA is shown in Figure 25-4 and described in Table 25-8.

Return to the Summary Table.

DAC Value Register - Active

Figure 25-4. DACVALA Register

15	14	13	12	11	10	9	8
RESERVED				DACVALA			
R-0h				R-0h			
7	6	5	4	3	2	1	0
DACVALA							
R-0h							

Table 25-8. DACVALA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DACVALA	R	0h	Active output code currently driven by the DAC Reset type: SYSRSn

25.5.2.4 DACVALS Register (Offset = 6h) [Reset = 0000h]

DACVALS is shown in Figure 25-5 and described in Table 25-9.

Return to the Summary Table.

DAC Value Register - Shadow

Figure 25-5. DACVALS Register

15	14	13	12	11	10	9	8
RESERVED				DACVALS			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
DACVALS							
R/W-0h							

Table 25-9. DACVALS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DACVALS	R/W	0h	Shadow output code to be loaded into DACVALA Reset type: SYSRSn

25.5.2.5 DACOUTEN Register (Offset = 8h) [Reset = 0000h]

DACOUTEN is shown in Figure 25-6 and described in Table 25-10.

Return to the Summary Table.

DAC Output Enable Register

Figure 25-6. DACOUTEN Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							DACOUTEN
R-0h							R/W-0h

Table 25-10. DACOUTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R	0h	Reserved
0	DACOUTEN	R/W	0h	DAC output enable 0 DAC output is disabled 1 DAC output is enabled Reset type: SYSRSn

25.5.2.6 DACLOCK Register (Offset = Ah) [Reset = 0000h]

DACLOCK is shown in Figure 25-7 and described in Table 25-11.

Return to the Summary Table.

DAC Lock Register

Figure 25-7. DACLOCK Register

15	14	13	12	11	10	9	8
KEY				RESERVED			
R-0/W-0h				R-0h			
7	6	5	4	3	2	1	0
RESERVED					DACOUTEN	DACVAL	DACCTL
R-0h					R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 25-11. DACLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	KEY	R-0/W	0h	Writes to this register succeed only if this field is written with a value of 0xA. Only 16-bit writes will succeed (provided the KEY matches). Read-modify-writes to individual bits in this register will be ignored. Reset type: SYSRSn
11-3	RESERVED	R	0h	Reserved
2	DACOUTEN	R/WSonce	0h	Lock write-access to the DACOUTEN register. 0 DACOUTEN register is not locked. Write 0 to this bit has no effect. 1 DACOUTEN register is locked. Only a system reset can clear this bit. Reset type: SYSRSn
1	DACVAL	R/WSonce	0h	Lock write-access to the DACVALS register. 0 DACVALS register is not locked. Write 0 to this bit has no effect. 1 DACVALS register is locked. Only a system reset can clear this bit. Reset type: SYSRSn
0	DACCTL	R/WSonce	0h	Lock write-access to the DACCTL register. 0 DACCTL register is not locked. Write 0 to this bit has no effect. 1 DACCTL register is locked. Only a system reset can clear this bit. Reset type: SYSRSn

25.5.2.7 DACTRIM Register (Offset = Ch) [Reset = 0000h]

DACTRIM is shown in Figure 25-8 and described in Table 25-12.

Return to the Summary Table.

DAC Trim Register

Figure 25-8. DACTRIM Register

15	14	13	12	11	10	9	8
RESERVED				RESERVED			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
OFFSET_TRIM							
R/W-0h							

Table 25-12. DACTRIM Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-8	RESERVED	R/W	0h	Reserved
7-0	OFFSET_TRIM	R/W	0h	DAC Offset Trim. This register should not be modified unless specifically indicated by TI Errata or other documentation. Modifying the contents of this register could cause this module to operate outside of datasheet specifications. Reset type: SYSRSn

Chapter 26

Comparator Subsystem (CMPSS)

The Comparator Subsystem (CMPSS) consists of analog comparators and supporting circuits that are useful for power applications such as peak current mode control, switched-mode power, power factor correction, voltage trip monitoring, and so forth.

See the *C2000 Real-Time Control Peripheral Reference Guide* for a list of all devices with modules of the same type, to determine the differences between the types, and for a list of device-specific differences within a type.

26.1 Introduction

The comparator subsystem is built around a number of modules. Each subsystem contains two comparators, two reference 12-bit DACs, and two digital filters. The subsystem also includes two ramp generators. The ramp generators ramp up and down.Comparators are denoted "H" or "L" within each module where “H” and “L” represent high and low, respectively. Each comparator generates a digital output which indicates whether the voltage on the positive input is greater than the voltage on the negative input. The positive input of the comparator is driven from an external pin (see the *Analog Subsystem* chapter for mux options available to the CMPSS). The negative input can be driven by an external pin or by the programmable reference 12-bit DAC . Each comparator output passes through a programmable digital filter that can remove spurious trip signals. An unfiltered output is also available if filtering is not required.

Two ramp generator circuits are optionally available to control the reference 12-bit DAC values for the high and low comparators in the subsystem. The DAC along with a wrapper can be used to generate a ramp which is used for slope compensation in Peak Current Mode Control (PCMC) and other applications. The subsystem also works with the EPWM to support Diode Emulation Mode.

26.1.1 CMPSS Related Collateral

Foundational Materials

- C28x Academy - CMPSS
- C29x Academy - Comparator Subsystem (CMPSS)
- *Real-Time Control Reference Guide*
  - Refer to the Comparator section

Expert Materials

- *Peak Current Control Realization for Boost Circuit Based On C2000™ MCU Application Report*
- *Peak Current Mode Controlled PSFB Converter Reference Design Using C2000™ Real-time MCU*
- *Understanding and Applying Current-Mode Control Theory Application Report*

26.1.2 Features

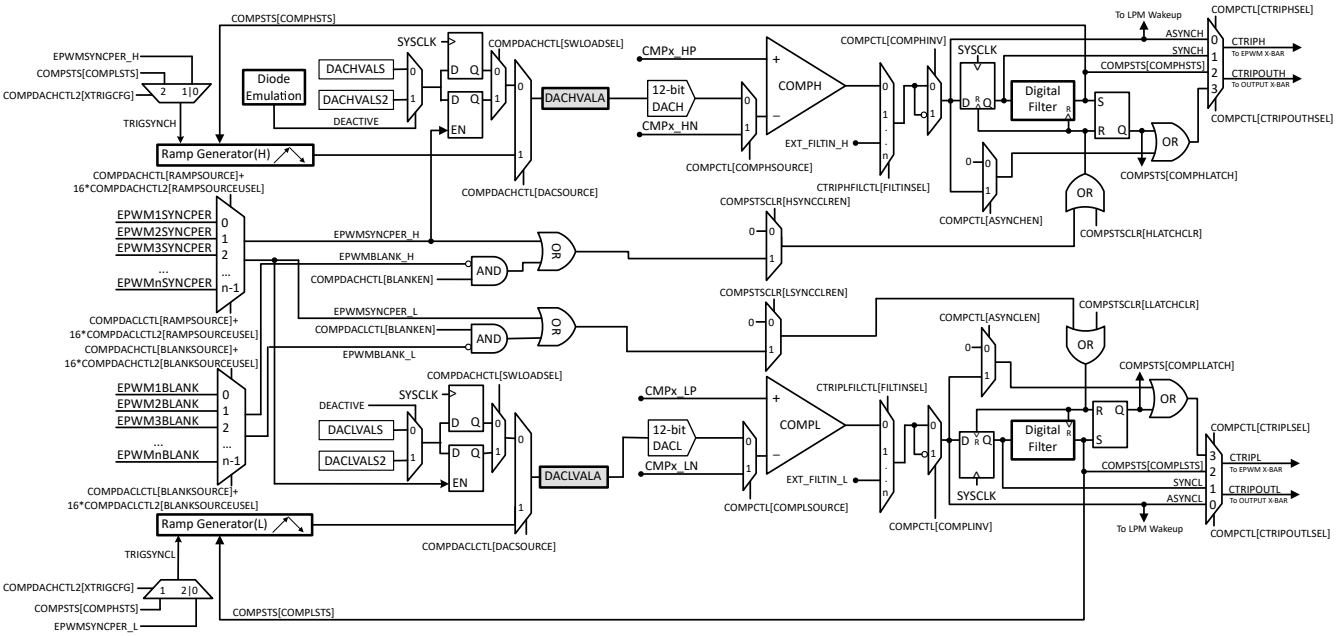
Each CMPSS includes:

- Two analog comparators
- Two independently programmable reference 12-bit DACs
- Dual decrementing/incrementing ramp generators
- Two digital filters, max filter clock prescale = 2^{24}
- Ability to synchronize submodules with EPWMSYNCPER
- Ability to synchronize output with SYSCLK
- Ability to latch output
- Ability to invert output
- Option to use hysteresis on the input
- Option for negative input of comparator to be driven by an external signal or by the reference DAC
- The DAC reference voltage can be set to VDDA or VDAC
- External connection to CMPSS filters
- Diode emulation support
- Ramp generator prescaler
- CMPSS Type-6 supports connection with ePWM Type-5
- CMPSS based Low Power Mode (LPM) wakeup

26.1.3 Block Diagram

The block diagram for the CMPSS is shown in Figure 26-1.

- CTRIPx(x= "H" or "L") signals are connected to the ePWM X-BAR for ePWM trip response. See the *Enhanced Pulse Width Modulator (ePWM)* chapter for more details on the ePWM X-BAR mux configuration.
- CTRIPxOUTx(x= "H" or "L") signals are connected to the Output X-BAR for external signaling. See the *General-Purpose Input/Output (GPIO)* chapter for more details on the Output X-BAR mux configuration.



Note

CMPxDACOUTEN only exists for the CMPSS 1 module on this device (CMP1LDACOUTEN in the CMPSSCTL register)

Figure 26-1. CMPSS Module Block Diagram

26.2 Comparator

The comparator generates a high digital output when the voltage on the positive input is greater than the voltage on the negative input, and a low digital output when the voltage on the positive input is less than the voltage on the negative input. The comparator is illustrated in Figure 26-2.

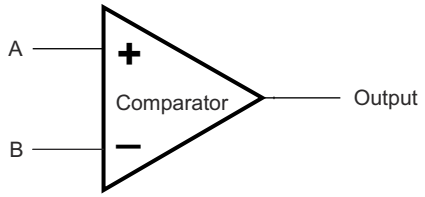


Figure 26-2. Comparator Block Diagram

Voltages	Output
Voltage A > Voltage B	1
Voltage A < Voltage B	0

26.3 Reference DAC

Each reference 12-bit DAC can be configured to drive a reference voltage into the negative input of the respective comparator. Some CMPSS instances also allow the low DAC output to be routed to a pin to act as an external DAC. In this case, all other CMPSS module functionality is not useable, including the high DAC, both comparators, ramp generation, and the digital filters.

Two sets of DACxVAL registers, DACxVALA and DACxVALS, are present for each reference 12-bit DAC. DACxVALA is a read-only register that actively controls the reference 12-bit DAC value. DACxVALS is a writable shadow register that loads into DACxVALA either immediately or synchronized with the next EPWMSYNCPER event. The high and low reference 12-bit DAC (DACx) can optionally source the register DACxVALA value from the ramp generator instead of the register DACxVALS.

The operating range of the reference 12-bit DAC is bounded by DACREF and VSSA. The high-voltage reference is VDDA by default, but the high voltage reference can be configured to be VDAC using the COMPDACHCTL register. The reference 12-bit DAC is illustrated in Figure 26-3.

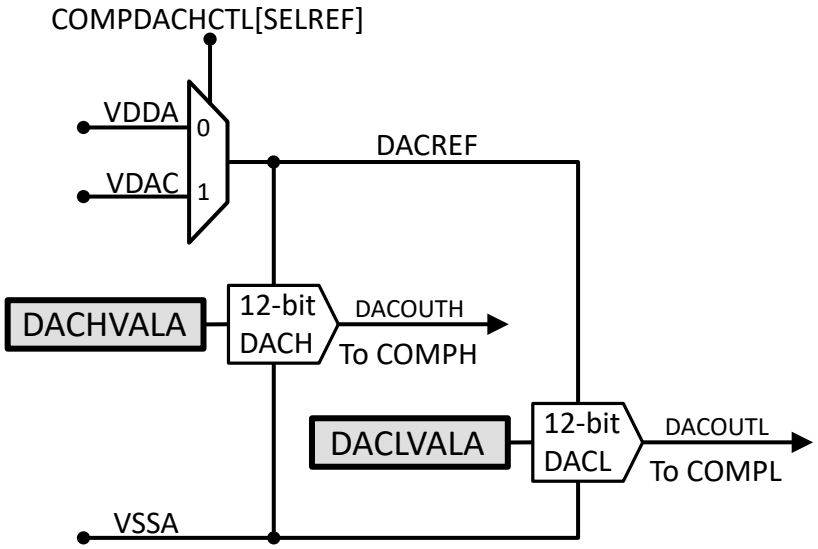


Figure 26-3. Reference DAC Block Diagram

The output of the reference 12-bit DAC can be calculated as:

$$DACOUT = \frac{(1 + DACVALA)*DACREF}{4096} \quad (21)$$

Note

- In the situations where both the DACH and DACL are driving the high and low comparators, a trip on one comparator can temporarily disturb the DAC output of the other comparator. The amount and length of time of this disturbance is specified in the device data sheet as “CMPSS DAC output disturbance” and “CMPSS DAC disturbance time”, respectively.

  Users must design the system carefully so that if the input signal crosses either DACH or DACL and trips the associated comparator, the input signal stays more than a “CMPSS DAC output disturbance” away from the other comparator trip point for “CMPSS DAC disturbance time”.
- The DACH setting must always be higher than the DACL setting. If the user is not using the DACL, the DACLVALS register must be set to 0 to avoid the comparator COMPL from tripping and affecting the DACH. In this case, there is no limitation on the DACHVALS setting. Accordingly, when not using the DACH, the user must set the DACHVALS register to the maximum.
- The CMPSS instance can be enabled before programming the reference DAC values.

26.4 Ramp Generator

This section discusses the characteristics and behavior of the ramp generator.

26.4.1 Ramp Generator Overview

The ramp generator produces a falling or rising ramp input for the high-reference 12-bit DAC and the low-reference 12-bit DAC when selected. In this mode, the reference 12-bit DAC uses the most-significant 12 bits of the RAMPSTS countdown register as the input. The low 4 bits of the RAMPSTS countdown register effectively act as a prescale for the falling or rising ramp rate configurable with RAMPxSTEPVALA. There is an additional dedicated prescaler for the ramp generator configurable with the RAMPCLKDIV register.

The ramp generator is enabled by setting DACSOURCE = 1. When DACSOURCE = 1 is selected, the value of RAMPSTS is loaded from RAMPxREFS and the register remains static until the selected TRIGSYNC signal is received. After receiving the selected TRIGSYNC signal, the value of RAMPxSTEPVALA is subtracted from RAMPSTS on every subsequent SYSCLK cycle.

To prevent the subtraction from commencing a SYSCLK cycle after a TRIGSYNC event, the RAMPDLYA register that serves as a delay counter can be used to hold off the RAMPSTS subtraction or addition. On receiving a TRIGSYNC event, the value of RAMPDLYA is decremented by one on every SYSCLK cycle until the register reaches zero. So, the RAMPSTS subtraction only begins when RAMPDLYA is zero.Similarly, in increment mode (RAMPDIR = 1), the RAMPSTS addition only begins when RAMPDLYA is zero.

Note

- For the ramp decrement mode of operation, set the registers COMPXINV and RAMPDIR to 0, and for the ramp increment mode of operation, set the registers COMPXINV and RAMPDIR to 1.

26.4.2 Ramp Generator Behavior

The ramp generator makes state changes on every rising edge of DACSOURCE, TRIGSYNC, and COMPHSTS.

On the rising edge of DACSOURCE: the registers RAMPxREFA, RAMPxSTEPVALA, and RAMPDLYA are loaded with the shadow registers. Also, the register RAMPSTS is loaded with RAMPxREFS.

On the rising edge of the selected TRIGSYNC: the registers RAMPxREFA, RAMPxSTEPVALA, and RAMPDLYA are loaded with the shadow registers. Also, the register RAMPSTS is loaded with RAMPxREFS and starts decrementing in the decrement mode (RAMPDIR = 0) or incrementing in the increment mode (RAMPDIR = 1) when RAMPDLYA counter reaches zero.

On the rising edge of COMPHSTS with RAMPLOADSEL = 1: the registers RAMPxREFA, RAMPxSTEPVALA, and RAMPDLYA are loaded with the shadow registers. Also, the register RAMPSTS is loaded with RAMPxREFS and stops decrementing in the decrement mode (RAMPDIR = 0) or incrementing in the increment mode (RAMPDIR = 1).

On the rising edge of COMPHSTS with RAMPLOADSEL = 0: the register RAMPSTS is loaded with RAMPxREFA. So, the register RAMPSTS stops decrementing and incrementing in the decrement mode and the increment mode, respectively.

Additionally, if the value of RAMPSTS reaches zero and the ramp generator is in the decrement mode (RAMPDIR = 0), the RAMPSTS register remains static at zero until the next TRIGSYNC is received. If the ramp generator is in the increment mode (RAMPDIR = 1) and the value of RAMPSTS reaches 0xFFFF, the RAMPSTS register value remains at that value until the next TRIGSYNC is received. These state changes are illustrated in the ramp generator block diagram in Figure 26-4.

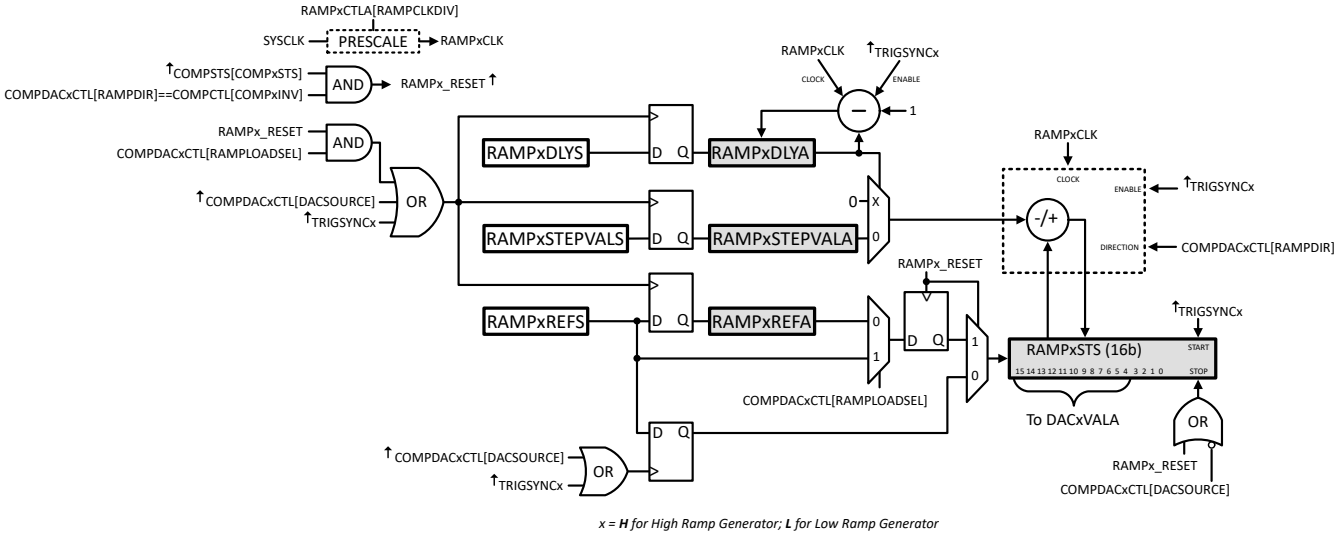


Figure 26-4. Ramp Generator Block Diagram

26.4.3 Ramp Generator Behavior at Corner Cases

Since the ramp generator makes state changes on every rising edge of TRIGSYNCx and COMPHSTS, the following behavior can be expected on instances when these two events occur simultaneously or very close together.

Case 1: COMPHSTS rising edge occurs one or more cycles before TRIGSYNC rising edge.
RAMPSTS stops decrementing on COMPHSTS rising edge event. RAMPSTS starts decrementing on TRIGSYNCx rising edge event when RAMPDLYA reaches 0.

Case 2: COMPHSTS rising edge occurs simultaneously as TRIGSYNCx rising edge.
EPWMSYNCPER rising edge event takes precedence and RAMPSTS starts decrementing when RAMPDLYA reaches 0. COMPHSTS rising edge event is ignored and does not halt RAMPSTS.

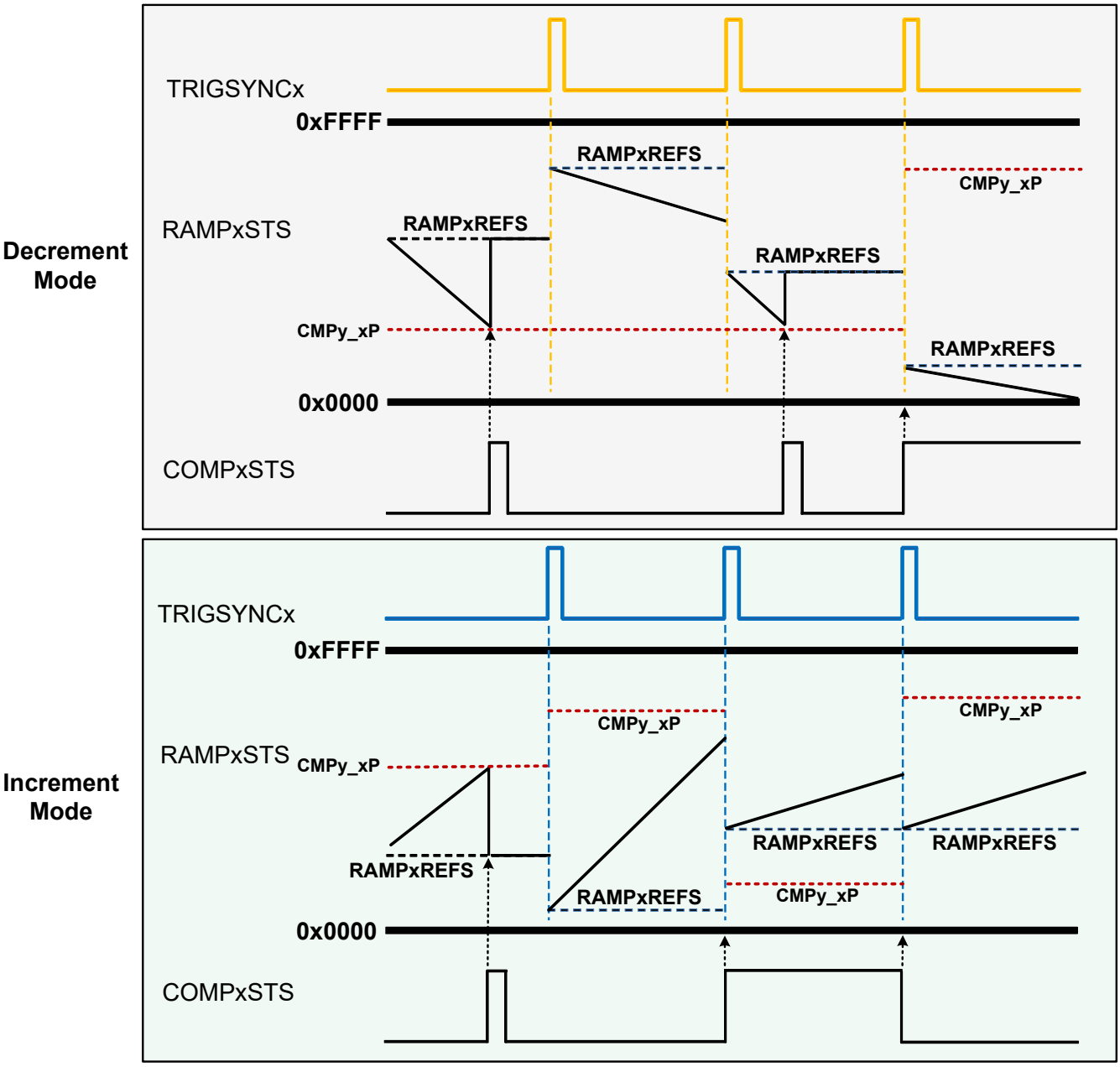
Case 3: COMPHSTS rising edge occurs one or more cycles after TRIGSYNCx rising edge but before RAMPDLYA reaches 0.
RAMPSTS does not decrement when RAMPDLYA reaches 0.

Case 4: COMPHSTS rising edge occurs simultaneously as RAMPDLYA reaches 0 from TRIGSYNCx rising edge.
RAMPSTS does not decrement.

This behavior is also illustrated in Figure 26-5.

Note

For the ramp decrement mode of operation, set the registers COMPXINV and RAMPDIR to 0, and for the ramp increment mode of operation, set the registers COMPXINV and RAMPDIR to 1.



x = H for High Ramp Generator; L for Low Ramp Generator
y= CMPSS module number

Figure 26-5. Ramp Generator Behavior

26.5 Digital Filter

The digital filter works on a window of FIFO samples (SAMPWIN) taken from the input. The filter output resolves to the majority value of the sample window, where majority is defined by the threshold (THRESH) value. If the majority threshold is not satisfied, the filter output remains unchanged.

For proper operation, the value of THRESH must be greater than SAMPWIN / 2 and less than or equal to SAMPWIN.

A prescale function (CLKPRESCALE) determines the filter sampling rate, where the filter FIFO captures one sample every prescale system clocks. Old data from the FIFO is discarded.

Note that for SAMPWIN, THRESH and CLKPRESCALE, the internal number used by the digital filter is + 1 in all cases. In essence, samples = SAMPWIN + 1, threshold = THRESH + 1 and prescale = CLKPRESCALE + 1.

A conceptual model of the digital filter is shown in Figure 26-6.

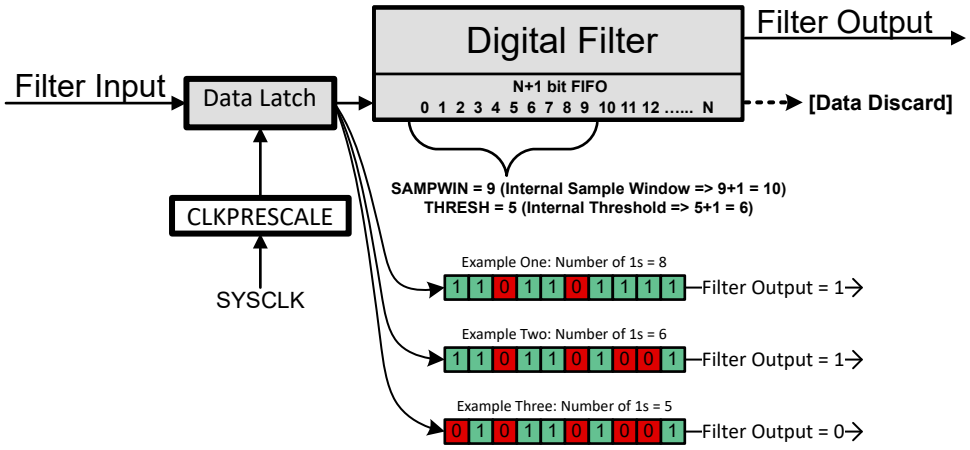


Figure 26-6. Digital Filter Behavior

Equivalent C code of the filter implementation is:

```
if (FILTER_OUTPUT == 0) {
    if (Num_1s_in_SAMPWIN >= THRESH) {
        FILTER_OUTPUT = 1;
    }
}
else {
    if (Num_0s_in_SAMPWIN >= THRESH) {
        FILTER_OUTPUT = 0;
    }
}
```

26.5.1 Filter Initialization Sequence

For proper operation of the digital filter, the following initialization sequence is recommended:

1. Configure and enable the comparator for operation.
2. Configure the digital filter parameters for operation:
   - Set SAMPWIN for the number of samples to monitor in the FIFO window.
   - Set THRESH for the threshold required for majority qualification.
   - Set CLKPRESCALE for the digital filter clock prescale value.
3. Initialize the sample values in the digital FIFO window by setting FILINIT.
4. Clear COMPSTS latch using COMPSTSCLR, if the latched path is desired.
5. Configure the CTRIP and CTRIPOUT signal paths.
6. If desired, configure the destination module, for example, ePWM, GPIO, and so on to accept the filtered signals.

26.6 Using the CMPSS

26.6.1 LATCHCLR, EPWMSYNCPER, and EPWMBLANK Signals

The LATCHCLR signal holds the digital filter, synchronization block, and the latch output in reset (0) after the required delays. The LATCHCLR signal is activated in software using xLATCHCLR (x = H or L). The LATCHCLR signal can also be activated by EPWMSYNCPER when xSYNCCLREN (x = H or L) is set. If a longer LATCHCLR signal is required, the EPWMBLANK signal can be used to extend the LATCHCLR signal by setting BLANKEN.

EPWMSYNCPER and EPWMBLANK (BLANKWDW) come from the Time-Base and Digital Compare submodules of the EPWM, respectively. For a detailed description of how these two signals are generated, refer to the respective submodule section in the *Enhanced Pulse Width Modulator (ePWM)* chapter.

The EPWMSYNCPER signal that loads DACxVALA when COMPDACCTL[SWLOADSEL] = 1 is a level trigger load. If TBCTR and TBPRD of the EPWM are both 0, EPWMSYNCPER is held at level high and DACxVALA is loaded immediately from DACxVALS irrespective of the value of COMPDACCTL[SWLOADSEL]. Due to this, configure the EPWM first before setting COMPDACCTL[SWLOADSEL] to 1.

Note

The name of the sync signal that the CMPSS receives from the EPWM has been updated from PWMSYNC to EPWMSYNCPER (SYNCPER/PWMSYNCPER/EPWMxSYNCPER) to avoid confusion with the other EPWM sync signals EPWMSYNCI and EPWMSYNCO. For a description of what are these signals, see the *Enhanced Pulse Width Modulator (ePWM)* chapter.

Note

26.6.2 Synchronizer, Digital Filter, and Latch Delays

The synchronization block adds a delay of 1-2 sysclks. If the digital filter is bypassed (all filter settings are 0), the digital filter adds a delay of 2 sysclks. The latch adds 1 sysclk delay.

26.6.3 Calibrating the CMPSS

The CMPSS has two sources of offset errors: comparator offset error and compdac offset error. In the data sheet, the comparator offset error is referred to as ***Input referred offset error*** and compdac offset error is referred to as ***Static offset error*** . See the device data sheet for the values.

If both inputs of the comparator are driven from a pin, only the comparator offset error applies. However if the inverting input of the comparator is driven from the compdac, then only the compdac offset error applies. This is because the compdac offset error includes comparator offset error.

Due to the offset errors, the CMPSS must be calibrated to make sure trips happen at the expected levels. The following flow outlines how the calibration can be performed if the inverting input of the comparator is driven from the compdac.

Notes before calibration:

1. A static DC signal is required on the non-inverting input of the comparator.
2. Hysteresis can be disabled for calibration and can be re-enabled after calibration is complete.
3. A noisy input can make calibration difficult, so use the latch with non-zero filter settings depending on how noisy is the signal on the non-inverting input.

This approach sweeps down the compdac:

1. Set the starting compdac value to max, 0xFFF.
   - Optional: Instead of setting the starting compdac value to maximum, set to ***Vtarget*** + ***Static offset error*** + ***Margin*** . Where ***Vtarget*** is the approximate DC voltage on the non-inverting input, ***Static offset error*** is the compdac offset error specification and ***Margin*** is some amount of guard band. This can lead to a faster calibration but only works if ***Vtarget*** is known. Alternatively, if ***Vtarget*** is unknown, the ADC can be used to convert ***Vtarget*** .
2. Decrement compdac value by 1.
3. Wait for compdac to settle.
4. Clear latch.
5. Wait for possible latch set.
6. If latch is set, trip code is found exit.
   - Optional: The trip code can be double checked by:
     a. Increasing compdac value by 1.
     b. Clear latch.
     c. Wait for possible latch set.
     d. Latch can be unset.
7. If latch is unset, go back to step 2 and repeat.

It is also possible to calibrate the CMPSS, if both inputs of the comparator are driven from a pin. For this case, the flow stays the same but the voltage on the inverting pin of the comparator is swept externally.

26.6.4 Enabling and Disabling the CMPSS Clock

If the clock to the CMPSS module is disabled while the comparator is active, the following behavior can be expected:

- The comparator remains unaffected and continues to trip from voltages on the inputs.
- If the reference 12-bit DAC is driving the negative input of the comparator, the voltage on the negative input remains static and unaffected but DACVALA can no longer be updated from the ramp generator or DACVALS.
- The ramp generator, synchronize block and digital filter freeze on the current states.

Enabling the clock to the CMPSS restores the clock to the state before the clock was disabled.

26.7 Software

26.7.1 CMPSS Registers to Driverlib Functions

Table 26-1. CMPSS Registers to Driverlib Functions

File	Driverlib Function
COMPCTL	
cmpss.h	CMPSS_enableModule
cmpss.h	CMPSS_disableModule
cmpss.h	CMPSS_configHighComparator
cmpss.h	CMPSS_configLowComparator
cmpss.h	CMPSS_configOutputsHigh
cmpss.h	CMPSS_configOutputsLow
COMPHYSCTL	
cmpss.h	CMPSS_setHysteresis
COMPSTS	
cmpss.c	CMPSS_configLatchOnPWMSYNC
cmpss.h	CMPSS_getStatus
cmpss.h	CMPSS_clearFilterLatchHigh
cmpss.h	CMPSS_clearFilterLatchLow
cmpss.h	CMPSS_enableLatchResetOnPWMSYNCHigh
cmpss.h	CMPSS_disableLatchResetOnPWMSYNCHigh
cmpss.h	CMPSS_enableLatchResetOnPWMSYNCLow
cmpss.h	CMPSS_disableLatchResetOnPWMSYNCLow
COMPSTSCLR	
cmpss.c	CMPSS_configLatchOnPWMSYNC
cmpss.h	CMPSS_clearFilterLatchHigh
cmpss.h	CMPSS_clearFilterLatchLow
cmpss.h	CMPSS_enableLatchResetOnPWMSYNCHigh
cmpss.h	CMPSS_disableLatchResetOnPWMSYNCHigh
cmpss.h	CMPSS_enableLatchResetOnPWMSYNCLow
cmpss.h	CMPSS_disableLatchResetOnPWMSYNCLow
COMPDACHCTL	
cmpss.c	CMPSS_configRamp
cmpss.c	CMPSS_configRampHigh
cmpss.c	CMPSS_configRampLow
cmpss.h	CMPSS_configDAC
cmpss.h	CMPSS_configDACHigh
cmpss.h	CMPSS_setRampDirectionHigh
cmpss.h	CMPSS_configureRampXTriggerHigh
cmpss.h	CMPSS_configureSyncSourceHigh
cmpss.h	CMPSS_configBlanking
cmpss.h	CMPSS_enableBlanking
cmpss.h	CMPSS_disableBlanking
cmpss.h	CMPSS_configBlankingSourceHigh
cmpss.h	CMPSS_enableBlankingHigh
cmpss.h	CMPSS_disableBlankingHigh
cmpss.h	CMPSS_enableDEmode

Table 26-1. CMPSS Registers to Driverlib Functions (continued)

File	Driverlib Function
cmpss.h	CMPSS_disableDEmode
cmpss.h	CMPSS_selectDEACTIVESource
cmpss.h	CMPSS_selectBlankSourceGroupHigh
cmpss.h	CMPSS_selectRampSourceGroupHigh
COMPDACHCTL2	
cmpss.h	CMPSS_configureRampXTriggerHigh
cmpss.h	CMPSS_enableDEmode
cmpss.h	CMPSS_disableDEmode
cmpss.h	CMPSS_selectDEACTIVESource
cmpss.h	CMPSS_selectBlankSourceGroupHigh
cmpss.h	CMPSS_selectRampSourceGroupHigh
DACHVALS	
cmpss.h	CMPSS_setDACValueHigh
cmpss.h	CMPSS_configHighDACShadowValueDE
DACHVALA	
cmpss.h	CMPSS_getDACValueHigh
RAMPHREFA	
cmpss.h	CMPSS_getMaxRampValue
cmpss.h	CMPSS_getRampReferenceHigh
RAMPHREFS	
cmpss.c	CMPSS_configRamp
cmpss.h	CMPSS_setMaxRampValue
cmpss.h	CMPSS_setRampReferenceHigh
RAMPHSTEPVALA	
cmpss.h	CMPSS_getRampDecValue
cmpss.h	CMPSS_getRampStepHigh
RAMPHCTLA	
cmpss.h	CMPSS_getRampClockDividerHigh
RAMPHSTEPVALS	
cmpss.c	CMPSS_configRamp
cmpss.h	CMPSS_setRampDecValue
cmpss.h	CMPSS_setRampStepHigh
RAMPHCTLS	
cmpss.h	CMPSS_setRampClockDividerHigh
RAMPHSTS	
-	
DACLVALS	
cmpss.h	CMPSS_setDACValueLow
cmpss.h	CMPSS_configLowDACShadowValueDE
DACLVALA	
cmpss.h	CMPSS_getDACValueLow
RAMPHDLYA	
cmpss.h	CMPSS_getRampDelayValue
cmpss.h	CMPSS_getRampDelayHigh
RAMPHDLYS	

Table 26-1. CMPSS Registers to Driverlib Functions (continued)

File	Driverlib Function
cmpss.c	CMPSS_configRamp
cmpss.h	CMPSS_setRampDelayValue
cmpss.h	CMPSS_setRampDelayHigh
CTRIPLFILCTL	
cmpss.c	CMPSS_configFilterLow
cmpss.h	CMPSS_initFilterLow
cmpss.h	CMPSS_configureFilterInputLow
CTRIPLFILCLKCTL	
cmpss.c	CMPSS_configFilterLow
CTRIPHFILCTL	
cmpss.c	CMPSS_configFilterHigh
cmpss.h	CMPSS_initFilterHigh
cmpss.h	CMPSS_configureFilterInputHigh
CTRIPHFILCLKCTL	
cmpss.c	CMPSS_configFilterHigh
COMPLOCK	
-	
DACHVALS2	
cmpss.h	CMPSS_configHighDACShadowValueDE
DACLVALS2	
cmpss.h	CMPSS_configLowDACShadowValueDE
COMPDACLCTL	
cmpss.h	CMPSS_configDACLow
cmpss.h	CMPSS_setRampDirectionLow
cmpss.h	CMPSS_configureSyncSourceLow
cmpss.h	CMPSS_configBlankingSourceLow
cmpss.h	CMPSS_enableBlankingLow
cmpss.h	CMPSS_disableBlankingLow
cmpss.h	CMPSS_selectBlankSourceGroupLow
cmpss.h	CMPSS_selectRampSourceGroupLow
COMPDACLCTL2	
cmpss.h	CMPSS_selectBlankSourceGroupLow
cmpss.h	CMPSS_selectRampSourceGroupLow
RAMPLREFA	
cmpss.h	CMPSS_getRampReferenceLow
RAMPLREFS	
cmpss.h	CMPSS_setRampReferenceLow
RAMPLSTEPVALA	
cmpss.h	CMPSS_getRampStepLow
RAMPLCTLA	
cmpss.h	CMPSS_getRampClockDividerLow
RAMPLSTEPVALS	
cmpss.h	CMPSS_setRampStepLow
RAMPLCTLS	
cmpss.h	CMPSS_setRampClockDividerLow

Table 26-1. CMPSS Registers to Driverlib Functions (continued)

File	Driverlib Function
RAMPLSTS	
-	
RAMPLDLYA	
cmpss.h	CMPSS_getRampDelayLow
RAMPLDLYS	
cmpss.h	CMPSS_setRampDelayLow
CTRIPLFILCLKCTL2	
cmpss.c	CMPSS_configFilterLow
CTRIPHFILCLKCTL2	
cmpss.c	CMPSS_configFilterHigh

26.7.2 CMPSS Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/cmpss

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

26.7.2.1 CMPSS Asynchronous Trip - SINGLE_CORE

FILE: cmpss_ex1_asynch.c

This example enables the CMPSS1 COMPH comparator and feeds the asynchronous CTRIPOUTH signal to the GPIO4/OUTPUTXBAR3 pin and CTRIPH to GPIO1/EPWM1B.

CMPSS is configured to generate trip signals to trip the EPWM signals. CMPIN1P is used to give positive input and internal DAC is configured to provide the negative input. Internal DAC is configured to provide a signal at VDD/2. An EPWM signal is generated at GPIO1 and is configured to be tripped by CTRIPOUTH.

When a low input(VSS) is provided to CMPIN1P,
- Trip signal(GPIO4) output is low
- PWM1B(GPIO1) gives a PWM signal

When a high input(higher than VDD/2) is provided to CMPIN1P,
- Trip signal(GPIO4) output turns high
- PWM1B(GPIO1) gets tripped and outputs as high

External Connections
- Give input on CMPIN1P
- Outputs can be observed on GPIO4 and GPIO1 using an oscilloscope

Watch Variables
- None

26.7.2.2 CMPSS Digital Filter Configuration - SINGLE_CORE

FILE: cmpss_ex2_digital_filter.c

This example enables the CMPSS1 COMPH comparator and feeds the output through the digital filter to the GPIO4/OUTPUTXBAR3 pin.

CMPIN1P is used to give positive input and internal DAC is configured to provide the negative input. Internal DAC is configured to provide a signal at VDD/2.

When a low input(VSS) is provided to CMPIN1P,
- GPIO4 output is low

When a high input(higher than VDD/2) is provided to CMPIN1P,

- GPIO4 output turns high

26.8 CMPSS Registers

This Section describes the CMPSS Registers.

26.8.1 CMPSS Base Address Table

Table 26-2. CMPSS Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
CMPSS_REGS	CMPSS1_BASE	0x700E_0000	YES	YES	YES	YES	YES	YES	-	YES
CMPSS_REGS	CMPSS2_BASE	0x700E_1000	YES	YES	YES	YES	YES	YES	-	YES
CMPSS_REGS	CMPSS3_BASE	0x700E_2000	YES	YES	YES	YES	YES	YES	-	YES
CMPSS_REGS	CMPSS4_BASE	0x700E_3000	YES	YES	YES	YES	YES	YES	-	YES
CMPSS_REGS	CMPSS5_BASE	0x700E_4000	YES	YES	YES	YES	YES	YES	-	YES
CMPSS_REGS	CMPSS6_BASE	0x700E_5000	YES	YES	YES	YES	YES	YES	-	YES
CMPSS_REGS	CMPSS7_BASE	0x700E_6000	YES	YES	YES	YES	YES	YES	-	YES
CMPSS_REGS	CMPSS8_BASE	0x700E_7000	YES	YES	YES	YES	YES	YES	-	YES
CMPSS_REGS	CMPSS9_BASE	0x700E_8000	YES	YES	YES	YES	YES	YES	-	YES
CMPSS_REGS	CMPSS10_BASE	0x700E_9000	YES	YES	YES	YES	YES	YES	-	YES
CMPSS_REGS	CMPSS11_BASE	0x700E_A000	YES	YES	YES	YES	YES	YES	-	YES
CMPSS_REGS	CMPSS12_BASE	0x700E_B000	YES	YES	YES	YES	YES	YES	-	YES

26.8.2 CMPSS_REGS Registers

Table 26-3 lists the memory-mapped registers for the CMPSS_REGS registers. All register offset addresses not listed in Table 26-3 should be considered as reserved locations and the register contents should not be modified.

Table 26-3. CMPSS_REGS Registers

Offset	Acronym	Register Name	Protection
0h	COMPCTL	CMPSS Comparator Control Register	
2h	COMPHYSCTL	CMPSS Comparator Hysteresis Control Register	
4h	COMPSTS	CMPSS Comparator Status Register	
6h	COMPSTSCLR	CMPSS Comparator Status Clear Register	
8h	COMPDACHCTL	CMPSS High DAC Control Register	
Ah	COMPDACHCTL2	CMPSS High DAC Control Register 2	
Ch	DACHVALS	CMPSS High DAC Value Shadow Register	
Eh	DACHVALA	CMPSS High DAC Value Active Register	
10h	RAMPHREFA	CMPSS High Ramp Reference Active Register	
14h	RAMPHREFS	CMPSS High Ramp Reference Shadow Register	
18h	RAMPHSTEPVALA	CMPSS High Ramp Step Value Active Register	
1Ah	RAMPHCTLA	CMPSS High Ramp Control Active Register	
1Ch	RAMPHSTEPVALS	CMPSS High Ramp Step Value Shadow Register	
1Eh	RAMPHCTLS	CMPSS High Ramp Control Shadow Register	
20h	RAMPHSTS	CMPSS High Ramp Status Register	
24h	DACLVALS	CMPSS Low DAC Value Shadow Register	
26h	DACLVALA	CMPSS Low DAC Value Active Register	
28h	RAMPHDLYA	CMPSS High Ramp Delay Active Register	
2Ah	RAMPHDLYS	CMPSS High Ramp Delay Shadow Register	
2Ch	CTRIPLFILCTL	CTRIPL Filter Control Register	
2Eh	CTRIPLFILCLKCTL	CTRIPL Filter Clock Control Register	
30h	CTRIPHFILCTL	CTRIPH Filter Control Register	
32h	CTRIPHFILCLKCTL	CTRIPH Filter Clock Control Register	
34h	COMPLOCK	CMPSS Lock Register	
38h	DACHVALS2	CMPSS High DAC Value Shadow Register 2	
3Ah	DACLVALS2	CMPSS Low DAC Value Shadow Register 2	
48h	COMPDACLCTL	CMPSS Low DAC Control Register	
4Ah	COMPDACLCTL2	CMPSS Low DAC Control Register 2	
50h	RAMPLREFA	CMPSS Low Ramp Reference Active Register	
54h	RAMPLREFS	CMPSS Low Ramp Reference Shadow Register	
58h	RAMPLSTEPVALA	CMPSS Low Ramp Step Value Active Register	
5Ah	RAMPLCTLA	CMPSS Low Ramp Control Active Register	
5Ch	RAMPLSTEPVALS	CMPSS Low Ramp Step Value Shadow Register	
5Eh	RAMPLCTLS	CMPSS Low Ramp Control Shadow Register	
60h	RAMPLSTS	CMPSS Low Ramp Status Register	
68h	RAMPLDLYA	CMPSS Low Ramp Delay Active Register	
6Ah	RAMPLDLYS	CMPSS Low Ramp Delay Shadow Register	
6Eh	CTRIPLFILCLKCTL2	CTRIPL Filter Clock Control Register 2	
72h	CTRIPHFILCLKCTL2	CTRIPH Filter Clock Control Register 2	

Complex bit access types are encoded to fit into small table cells. Table 26-4 shows the codes that are used for access types in this section.

Table 26-4. CMPSS_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value

26.8.2.1 COMPCTL Register (Offset = 0h) [Reset = 0000h]

COMPCTL is shown in Figure 26-7 and described in Table 26-5.

Return to the Summary Table.

CMPSS Comparator Control Register

Figure 26-7. COMPCTL Register

15	14	13	12	11	10	9	8
COMPDACE	ASYNCLEN	CTRIPOUTLSEL		CTRIPLSEL		COMPLINV	COMPLSOURCE
R/W-0h	R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	ASYNCHEN	CTRIPOUTHSEL		CTRIPHSEL		COMPHINV	COMPHSOURCE
R-0h	R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h

Table 26-5. COMPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	COMPDACE	R/W	0h	Comparator/DAC enable. 0 Comparator/DAC disabled 1 Comparator/DAC enabled Reset type: SYSRSn
14	ASYNCLEN	R/W	0h	Low comparator asynchronous path enable. Allows asynchronous comparator output to feed into OR gate with latched digital filter signal when CTRIPLSEL=3 or CTRIPOUTLSEL=3. 0 Asynchronous comparator output does not feed into OR gate with latched digital filter output 1 Asynchronous comparator output feeds into OR gate with latched digital filter output Reset type: SYSRSn
13-12	CTRIPOUTLSEL	R/W	0h	Low comparator CTRIPOUTL source select. 0 Asynchronous comparator output drives CTRIPOUTL 1 Synchronous comparator output drives CTRIPOUTL 2 Output of digital filter drives CTRIPOUTL 3 Latched output of digital filter drives CTRIPOUTL Reset type: SYSRSn
11-10	CTRIPLSEL	R/W	0h	Low comparator CTRIPL source select. 0 Asynchronous comparator output drives CTRIPL 1 Synchronous comparator output drives CTRIPL 2 Output of digital filter drives CTRIPL 3 Latched output of digital filter drives CTRIPL Reset type: SYSRSn
9	COMPLINV	R/W	0h	Low comparator output invert. 0 Output of comparator is not inverted 1 Output of comparator is inverted Reset type: SYSRSn
8	COMPLSOURCE	R/W	0h	Low comparator input source. 0 Inverting input of comparator driven by internal DAC 1 Inverting input of comparator driven through external pin Reset type: SYSRSn
7	RESERVED	R	0h	Reserved

Table 26-5. COMPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	ASYNCHEN	R/W	0h	High comparator asynchronous path enable. Allows asynchronous comparator output to feed into OR gate with latched digital filter signal when CTRIPHSEL=3 or CTRIPOUTHSEL=3. 0 Asynchronous comparator output does not feed into OR gate with latched digital filter output 1 Asynchronous comparator output feeds into OR gate with latched digital filter output Reset type: SYSRSn
5-4	CTRIPOUTHSEL	R/W	0h	High comparator CTRIPOUTH source select. 0 Asynchronous comparator output drives CTRIPOUTH 1 Synchronous comparator output drives CTRIPOUTH 2 Output of digital filter drives CTRIPOUTH 3 Latched output of digital filter drives CTRIPOUTH Reset type: SYSRSn
3-2	CTRIPHSEL	R/W	0h	High comparator CTRIPH source select. 0 Asynchronous comparator output drives CTRIPH 1 Synchronous comparator output drives CTRIPH 2 Output of digital filter drives CTRIPH 3 Latched output of digital filter drives CTRIPH Reset type: SYSRSn
1	COMPHINV	R/W	0h	High comparator output invert. 0 Output of comparator is not inverted 1 Output of comparator is inverted Reset type: SYSRSn
0	COMPHSOURCE	R/W	0h	High comparator input source. 0 Inverting input of comparator driven by internal DAC 1 Inverting input of comparator driven through external pin Reset type: SYSRSn

26.8.2.2 COMPHYSCTL Register (Offset = 2h) [Reset = 0000h]

COMPHYSCTL is shown in Figure 26-8 and described in Table 26-6.

Return to the Summary Table.

CMPSS Comparator Hysteresis Control Register

Figure 26-8. COMPHYSCTL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				COMPHYS			
R-0h				R/W-0h			

Table 26-6. COMPHYSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-0	COMPHYS	R/W	0h	Comparator hysteresis. Sets the amount of hysteresis on the comparator inputs. 0 None 1 Set to typical hysteresis 2 Set to 2x of typical hysteresis 3 Set to 3x of typical hysteresis 4 Set to 4x of typical hysteresis others : undefined Reset type: SYSRSn

26.8.2.3 COMPSTS Register (Offset = 4h) [Reset = 0000h]

COMPSTS is shown in Figure 26-9 and described in Table 26-7.

Return to the Summary Table.

CMPSS Comparator Status Register

Figure 26-9. COMPSTS Register

15	14	13	12	11	10	9	8
RESERVED						COMPLLATCH	COMPLSTS
R-0h						R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED						COMPHLATCH	COMPHSTS
R-0h						R-0h	R-0h

Table 26-7. COMPSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9	COMPLLATCH	R	0h	Latched value of low comparator digital filter output Reset type: SYSRSn
8	COMPLSTS	R	0h	Low comparator digital filter output Reset type: SYSRSn
7-2	RESERVED	R	0h	Reserved
1	COMPHLATCH	R	0h	Latched value of high comparator digital filter output Reset type: SYSRSn
0	COMPHSTS	R	0h	High comparator digital filter output Reset type: SYSRSn

26.8.2.4 COMPSTSCLR Register (Offset = 6h) [Reset = 0000h]

COMPSTSCLR is shown in Figure 26-10 and described in Table 26-8.

Return to the Summary Table.

CMPSS Comparator Status Clear Register

Figure 26-10. COMPSTSCLR Register

15	14	13	12	11	10	9	8
RESERVED					LSYNCCLREN	LLATCHCLR	RESERVED
R-0h					R/W-0h	R-0/W1S-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED					HSYNCCLREN	HLATCHCLR	RESERVED
R-0h					R/W-0h	R-0/W1S-0h	R-0h

Table 26-8. COMPSTSCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R	0h	Reserved
10	LSYNCCLREN	R/W	0h	Low comparator latch EPWMSYNCPER clear. Enable EPWMSYNCPER reset of low comparator digital filter output latch COMPSTS[COMPLLATCH]. 0 EPWMSYNCPER will not reset latch 1 EPWMSYNCPER will reset latch Reset type: SYSRSn
9	LLATCHCLR	R-0/W1S	0h	Low comparator latch software clear. Perform software reset of low comparator digital filter output latch COMPSTS[COMPLLATCH]. Reads always return 0. 0 No effect 1 Generate a single pulse of latch reset signal for COMPSTS[COMPLLATCH] Reset type: SYSRSn
8-3	RESERVED	R	0h	Reserved
2	HSYNCCLREN	R/W	0h	High comparator latch EPWMSYNCPER clear. Enable EPWMSYNCPER reset of high comparator digital filter output latch COMPSTS[COMPHLATCH]. 0 EPWMSYNCPER will not reset latch 1 EPWMSYNCPER will reset latch Reset type: SYSRSn
1	HLATCHCLR	R-0/W1S	0h	High comparator latch software clear. Perform software reset of high comparator digital filter output latch COMPSTS[COMPHLATCH]. Reads always return 0. 0 No effect 1 Generate a single pulse of latch reset signal for COMPSTS[COMPHLATCH] Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

26.8.2.5 COMPDACHCTL Register (Offset = 8h) [Reset = 0000h]

COMPDACHCTL is shown in Figure 26-11 and described in Table 26-9.

Return to the Summary Table.

CMPSS High DAC Control Register

Figure 26-11. COMPDACHCTL Register

15	14	13	12	11	10	9	8
FREESOFT		RAMPDIR	BLANKEN	BLANKSOURCE			
R/W-0h		R/W-0h	R/W-0h	R/W-0h			
7	6	5	4	3	2	1	0
SWLOADSEL	RAMPLOADSEL	SELREF	RAMPSOURCE				DACSOURCE
R/W-0h	R/W-0h	R/W-0h	R/W-0h				R/W-0h

Table 26-9. COMPDACHCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	FREESOFT	R/W	0h	Free-run or software-run emulation behavior. Behavior of the High/Low ramp generators during emulation suspend. 00b Ramp generator stops immediately during emulation suspend 01b Ramp generator completes current ramp and stops at next EPWMSYNCPER during emulation suspend 1Xb Ramp generator runs freely Reset type: SYSRSn
13	RAMPDIR	R/W	0h	High Ramp Generator Direction control bit. 0 Decrementing Ramp. 1 Incrementing Ramp. Reset type: SYSRSn
12	BLANKEN	R/W	0h	COMPH EPWMBLANK enable. This bit enables the EPWMBLANK signal. 0 EPWMBLANK signal is disabled. 1 EPWMBLANK signal is enabled. Reset type: SYSRSn
11-8	BLANKSOURCE	R/W	0h	COMPH EPWMBLANK source select. This bit field determines which EPWMnBLANK is passed on as the EPWMBLANK signal. Where n represents the maximum number of EPWMBLANK signals available on the device: 0 EPWM1BLANK 1 EPWM2BLANK 2 EPWM3BLANK ... n-1 EPWMnBLANK Reset type: SYSRSn
7	SWLOADSEL	R/W	0h	Software load select. Determines whether DACxVALA is updated from DACxVALS on SYSCLK or EPWMSYNCPER. 0 DACxVALA is updated from DACxVALS on SYSCLK 1 DACxVALA is updated from DACxVALS on EPWMSYNCPER Reset type: SYSRSn
6	RAMPLOADSEL	R/W	0h	Ramp load select. Determines whether RAMPHSTS is updated from RAMPHREFA or RAMPHREFS when COMPSTS[COMPHSTS] is triggered. 0 RAMPHSTS is loaded from RAMPHREFA 1 RAMPHSTS is loaded from RAMPHREFS Reset type: SYSRSn

Table 26-9. COMPDACHCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	SELREF	R/W	0h	DAC reference select. Determines which voltage supply is used as the reference for the internal comparator DACs. 0 VDDA is the voltage reference for the DAC 1 VDAC is the voltage reference for the DAC Reset type: SYSRSn
4-1	RAMPSOURCE	R/W	0h	High Ramp generator source select. Determines which EPWMSYNCPER signal is used within the COMPH Where n represents the maximum number of EPWMSYNCPER signals available on the device: 0 EPWM1SYNCPER 1 EPWM2SYNCPER 2 EPWM3SYNCPER ... n-1 EPWMnSYNCPER Reset type: SYSRSn
0	DACSOURCE	R/W	0h	DACH source select. Determines whether DACHVALA is updated from DACHVALS or from the high ramp generator. 0 DACHVALA is updated from DACHVALS 1 DACHVALA is updated from the high ramp generator Reset type: SYSRSn

26.8.2.6 COMPDACHCTL2 Register (Offset = Ah) [Reset = 0000h]

COMPDACHCTL2 is shown in Figure 26-12 and described in Table 26-10.

Return to the Summary Table.

CMPSS High DAC Control Register 2

Figure 26-12. COMPDACHCTL2 Register

15	14	13	12	11	10	9	8
RESERVED		XTRIGCFG		RESERVED	RAMPSOURCEUSEL	RESERVED	BLANKSOURCEUSEL
R-0h		R/W-0h		R-0h	R/W-0h	R-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED		DEACTIVESEL					DEENABLE
R-0h		R/W-0h					R/W-0h

Table 26-10. COMPDACHCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	RESERVED	R	0h	Reserved
13-12	XTRIGCFG	R/W	0h	Ramp Generator Cross Trigger Configuration 00 : RAMPH and RAMPL operate independently (RAMPH SOR and RAMPL SOR are triggered by their corresponding selected PWMSYNCx signals) 01 : RAMPL is cross triggered by RAMPH (RAMPH SOR is triggered by its selected PWMSYNCx signal and RAMPL SOR is triggered by RAMPH EOR) 10 : RAMPH is cross triggered by RAMPL (RAMPL SOR is triggered by its selected PWMSYNCx signal and RAMPH SOR is triggered by RAMPL EOR 11 : Reserved Note : RAMPy SOR = Start of Ramp, RAMPy EOR = End of Ramp (COMPySTS signal) XTRIGCFG[0] = XTRIGCFG-L XTRIGCFG[1] = XTRIGCFG-H Reset type: SYSRSn
11	RESERVED	R	0h	Reserved
10	RAMPSOURCEUSEL	R/W	0h	0: Selects EPWM1 to 16 as RAMP source for RAMPH 1: Selects EPWM17 to 32 as RAMP source for RAMPH Reset type: SYSRSn
9	RESERVED	R	0h	Reserved
8	BLANKSOURCEUSEL	R/W	0h	0: Selects EPWM1 to 16 as BLANK source for COMPH 1: Selects EPWM17 to 32 as BLANK source for COMPH Reset type: SYSRSn
7-6	RESERVED	R	0h	Reserved
5-1	DEACTIVESEL	R/W	0h	DEACTIVE source select. This bit field determines which EPWMn.DEACTIVE is passed on as the DEACTIVE signal. Where n represents the maximum number of EPWMDEACTIVE signals available on the device: 0 : EPWM1.DEACTIVE 1 : EPWM2.DEACTIVE 2 : EPWM3.DEACTIVE 3 : EPWM4.DEACTIVE .... n-1 : EPWMn.DEACTIVE Reset type: SYSRSn

Table 26-10. COMPDACHCTL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	DEENABLE	R/W	0h	Diode Emulation mode enable. 0 DE mode features disabled. 1 DE mode features enabled. Reset type: SYSRSn

26.8.2.7 DACHVALS Register (Offset = Ch) [Reset = 0000h]

DACHVALS is shown in Figure 26-13 and described in Table 26-11.

Return to the Summary Table.

CMPSS High DAC Value Shadow Register

Figure 26-13. DACHVALS Register

15	14	13	12	11	10	9	8
RESERVED				DACVAL			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
DACVAL							
R/W-0h							

Table 26-11. DACHVALS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DACVAL	R/W	0h	High DAC shadow value. When COMPDACCTL[DACSOURCE]=0, the value of DACHVALS is loaded into DACHVALA on the trigger signal selected by COMPDACCTL[SWLOADSEL]. Reset type: SYSRSn

26.8.2.8 DACHVALA Register (Offset = Eh) [Reset = 0000h]

DACHVALA is shown in Figure 26-14 and described in Table 26-12.

Return to the Summary Table.

CMPSS High DAC Value Active Register

Figure 26-14. DACHVALA Register

15	14	13	12	11	10	9	8
RESERVED				DACVAL			
R-0h				R-0h			
7	6	5	4	3	2	1	0
DACVAL							
R-0h							

Table 26-12. DACHVALA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DACVAL	R	0h	High DAC active value. Value that is actively driven by the high DAC. Reset type: SYSRSn

26.8.2.9 RAMPHREFA Register (Offset = 10h) [Reset = 0000h]

RAMPHREFA is shown in Figure 26-15 and described in Table 26-13.

Return to the Summary Table.

CMPSS High Ramp Reference Active Register

Figure 26-15. RAMPHREFA Register

15	14	13	12	11	10	9	8
RAMPHREF							
R-0h							
7	6	5	4	3	2	1	0
RAMPHREF							
R-0h							

Table 26-13. RAMPHREFA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RAMPHREF	R	0h	High Ramp reference active value. Latched value to be loaded into ramp generator RAMHPSTS. Reset type: SYSRSn

26.8.2.10 RAMPHREFS Register (Offset = 14h) [Reset = 0000h]

RAMPHREFS is shown in Figure 26-16 and described in Table 26-14.

Return to the Summary Table.

CMPSS High Ramp Reference Shadow Register

Figure 26-16. RAMPHREFS Register

15	14	13	12	11	10	9	8
RAMPHREF							
R/W-0h							
7	6	5	4	3	2	1	0
RAMPHREF							
R/W-0h							

Table 26-14. RAMPHREFS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RAMPHREF	R/W	0h	High Ramp reference shadow. Unlatched value to be loaded into ramp generator RAMPHSTS. Reset type: SYSRSn

26.8.2.11 RAMPHSTEPVALA Register (Offset = 18h) [Reset = 0000h]

RAMPHSTEPVALA is shown in Figure 26-17 and described in Table 26-15.

Return to the Summary Table.

CMPSS High Ramp Step Value Active Register

Figure 26-17. RAMPHSTEPVALA Register

15	14	13	12	11	10	9	8
RAMPHSTEPVAL							
R-0h							
7	6	5	4	3	2	1	0
RAMPHSTEPVAL							
R-0h							

Table 26-15. RAMPHSTEPVALA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RAMPHSTEPVAL	R	0h	High Ramp step value active. Latched value that will be subtracted from RAMPHSTS. Reset type: SYSRSn

26.8.2.12 RAMPHCTLA Register (Offset = 1Ah) [Reset = 0000h]

RAMPHCTLA is shown in Figure 26-18 and described in Table 26-16.

Return to the Summary Table.

CMPSS High Ramp Control Active Register

Figure 26-18. RAMPHCTLA Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RAMPCLKDIV			
R-0h				R-0h			

Table 26-16. RAMPHCTLA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-0	RAMPCLKDIV	R	0h	Ramp High Clock Divider Active Value RAMPCLK = SYSCLK/(RAMPCLKDIV+1) Reset type: SYSRSn

26.8.2.13 RAMPHSTEPVALS Register (Offset = 1Ch) [Reset = 0000h]

RAMPHSTEPVALS is shown in Figure 26-19 and described in Table 26-17.

Return to the Summary Table.

CMPSS High Ramp Step Value Shadow Register

Figure 26-19. RAMPHSTEPVALS Register

15	14	13	12	11	10	9	8
RAMPHSTEPVAL							
R/W-0h							
7	6	5	4	3	2	1	0
RAMPHSTEPVAL							
R/W-0h							

Table 26-17. RAMPHSTEPVALS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RAMPHSTEPVAL	R/W	0h	High Ramp step value shadow. Unlatched value to be loaded into RAMPHSTEPVALA. Reset type: SYSRSn

26.8.2.14 RAMPHCTLS Register (Offset = 1Eh) [Reset = 0000h]

RAMPHCTLS is shown in Figure 26-20 and described in Table 26-18.

Return to the Summary Table.

CMPSS High Ramp Control Shadow Register

Figure 26-20. RAMPHCTLS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RAMPCLKDIV			
R-0h				R/W-0h			

Table 26-18. RAMPHCTLS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-0	RAMPCLKDIV	R/W	0h	Ramp High Clock Divider Shadow Value This will be the unlatched value that will be loaded into the RAMPCLKDIV field of the RAMPCTLA register Reset type: SYSRSn

26.8.2.15 RAMPHSTS Register (Offset = 20h) [Reset = 0000h]

RAMPHSTS is shown in Figure 26-21 and described in Table 26-19.

Return to the Summary Table.

CMPSS High Ramp Status Register

Figure 26-21. RAMPHSTS Register

15	14	13	12	11	10	9	8
RAMPHVALUE							
R-0h							
7	6	5	4	3	2	1	0
RAMPHVALUE							
R-0h							

Table 26-19. RAMPHSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RAMPHVALUE	R	0h	High Ramp value. Present value of ramp generator. Reset type: SYSRSn

26.8.2.16 DACLVALS Register (Offset = 24h) [Reset = 0000h]

DACLVALS is shown in Figure 26-22 and described in Table 26-20.

Return to the Summary Table.

CMPSS Low DAC Value Shadow Register

Figure 26-22. DACLVALS Register

15	14	13	12	11	10	9	8
RESERVED				DACVAL			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
DACVAL							
R/W-0h							

Table 26-20. DACLVALS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DACVAL	R/W	0h	Low DAC shadow value. value to be loaded into DACLVALA on the trigger signal selected by COMPDACCTL[SWLOADSEL]. Reset type: SYSRSn

26.8.2.17 DACLVALA Register (Offset = 26h) [Reset = 0000h]

DACLVALA is shown in Figure 26-23 and described in Table 26-21.

Return to the Summary Table.

CMPSS Low DAC Value Active Register

Figure 26-23. DACLVALA Register

15	14	13	12	11	10	9	8
RESERVED				DACVAL			
R-0h				R-0h			
7	6	5	4	3	2	1	0
DACVAL							
R-0h							

Table 26-21. DACLVALA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DACVAL	R	0h	Low DAC active value. Value that is actively driven by the low DAC. Reset type: SYSRSn

26.8.2.18 RAMPHDLYA Register (Offset = 28h) [Reset = 0000h]

RAMPHDLYA is shown in Figure 26-24 and described in Table 26-22.

Return to the Summary Table.

CMPSS High Ramp Delay Active Register

Figure 26-24. RAMPHDLYA Register

15	14	13	12	11	10	9	8
RESERVED			DELAY				
R-0h			R-0h				
7	6	5	4	3	2	1	0
DELAY							
R-0h							

Table 26-22. RAMPHDLYA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12-0	DELAY	R	0h	High Ramp delay active value. Latched value of the number of cycles to delay the start of the ramp generator stepper after a EPWMSYNCPER is received. Reset type: SYSRSn

26.8.2.19 RAMPHDLYS Register (Offset = 2Ah) [Reset = 0000h]

RAMPHDLYS is shown in Figure 26-25 and described in Table 26-23.

Return to the Summary Table.

CMPSS High Ramp Delay Shadow Register

Figure 26-25. RAMPHDLYS Register

15	14	13	12	11	10	9	8
RESERVED			DELAY				
R-0h			R/W-0h				
7	6	5	4	3	2	1	0
DELAY							
R/W-0h							

Table 26-23. RAMPHDLYS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12-0	DELAY	R/W	0h	High Ramp delay shadow value. Unlatched value to be loaded into RAMPHDLYA. Reset type: SYSRSn

26.8.2.20 CTRIPLFILCTL Register (Offset = 2Ch) [Reset = 0000h]

CTRIPLFILCTL is shown in Figure 26-26 and described in Table 26-24.

Return to the Summary Table.

CTRIPL Filter Control Register

Figure 26-26. CTRIPLFILCTL Register

15	14	13	12	11	10	9	8
FILINIT	THRESH						SAMPWIN
R-0/W1S-0h	R/W-0h						R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN						FILTINSEL	
R/W-0h						R/W-0h	

Table 26-24. CTRIPLFILCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	Low filter initialization. 0 No effect 1 Initialize all samples to the filter input value Reset type: SYSRSn
14-9	THRESH	R/W	0h	Low filter majority voting threshold. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Threshold used is THRESH+1. Reset type: SYSRSn
8-3	SAMPWIN	R/W	0h	Low filter sample window size. Number of samples to monitor is SAMPWIN+1. Reset type: SYSRSn
2-0	FILTINSEL	R/W	0h	Low filter Input Mux Select Bit 0 Selects the COMPL output as the filter input 1 Selects the external signal EXT_FILTIN_L[1] as the filter input 2 Selects the external signal EXT_FILTIN_L[2] as the filter input ... ... 7 Selects the external signal EXT_FILTIN_L[7] as the filter input Reset type: SYSRSn

26.8.2.21 CTRIPLFILCLKCTL Register (Offset = 2Eh) [Reset = 0000h]

CTRIPLFILCLKCTL is shown in Figure 26-27 and described in Table 26-25.

Return to the Summary Table.

CTRIPL Filter Clock Control Register

Figure 26-27. CTRIPLFILCLKCTL Register

15	14	13	12	11	10	9	8
CLKPRESCALE							
R/W-0h							
7	6	5	4	3	2	1	0
CLKPRESCALE							
R/W-0h							

Table 26-25. CTRIPLFILCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CLKPRESCALE	R/W	0h	Low filter sample clock prescale. Number of system clocks between samples is CLKPRESCALE+1. Reset type: SYSRSn

26.8.2.22 CTRIPHFILCTL Register (Offset = 30h) [Reset = 0000h]

CTRIPHFILCTL is shown in Figure 26-28 and described in Table 26-26.

Return to the Summary Table.

CTRIPH Filter Control Register

Figure 26-28. CTRIPHFILCTL Register

15	14	13	12	11	10	9	8
FILINIT	THRESH						SAMPWIN
R-0/W1S-0h	R/W-0h						R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN						FILTINSEL	
R/W-0h						R/W-0h	

Table 26-26. CTRIPHFILCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	High filter initialization. 0 No effect 1 Initialize all samples to the filter input value Reset type: SYSRSn
14-9	THRESH	R/W	0h	High filter majority voting threshold. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Threshold used is THRESH+1. Reset type: SYSRSn
8-3	SAMPWIN	R/W	0h	High filter sample window size. Number of samples to monitor is SAMPWIN+1. Reset type: SYSRSn
2-0	FILTINSEL	R/W	0h	High filter Input Mux Select Bit 0 Selects the COMPH output as the filter input 1 Selects the external signal EXT_FILTIN_H[1] as the filter input 2 Selects the external signal EXT_FILTIN_H[2] as the filter input ... ... 7 Selects the external signal EXT_FILTIN_H[7] as the filter input Reset type: SYSRSn

26.8.2.23 CTRIPHFILCLKCTL Register (Offset = 32h) [Reset = 0000h]

CTRIPHFILCLKCTL is shown in Figure 26-29 and described in Table 26-27.

Return to the Summary Table.

CTRIPH Filter Clock Control Register

Figure 26-29. CTRIPHFILCLKCTL Register

15	14	13	12	11	10	9	8
CLKPRESCALE							
R/W-0h							
7	6	5	4	3	2	1	0
CLKPRESCALE							
R/W-0h							

Table 26-27. CTRIPHFILCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CLKPRESCALE	R/W	0h	High filter sample clock prescale. Number of system clocks between samples is CLKPRESCALE+1. Reset type: SYSRSn

26.8.2.24 COMPLOCK Register (Offset = 34h) [Reset = 0000h]

COMPLOCK is shown in Figure 26-30 and described in Table 26-28.

Return to the Summary Table.

CMPSS Lock Register

Figure 26-30. COMPLOCK Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			RESERVED	CTRIP	DACCTL	COMPHYSCTL	COMPCTL
R-0h			R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 26-28. COMPLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R	0h	Reserved
4	RESERVED	R/WSonce	0h	Reserved
3	CTRIP	R/WSonce	0h	Lock write-access to the CTRIPxFILTCTL and CTRIPxFILCLKCTL* registers. 0 CTRIPxFILCTL and CTRIPxFILCLKCTL* registers are not locked. Write 0 to this bit has no effect. 1 CTRIPxFILCTL and CTRIPxFILCLKCTL* registers are locked. Only a system reset can clear this bit. Reset type: SYSRSn
2	DACCTL	R/WSonce	0h	Lock write-access to the COMPDAC*CTL* register(s). 0 COMPDAC*CTL* register(s) not locked. Write 0 to this bit has no effect. 1 COMPDAC*CTL* register(s) locked. Only a system reset can clear this bit. Reset type: SYSRSn
1	COMPHYSCTL	R/WSonce	0h	Lock write-access to the COMPHYSCTL register. 0 COMPHYSCTL register is not locked. Write 0 to this bit has no effect. 1 COMPHYSCTL register is locked. Only a system reset can clear this bit. Reset type: SYSRSn
0	COMPCTL	R/WSonce	0h	Lock write-access to the COMPCTL register. 0 COMPCTL register is not locked. Write 0 to this bit has no effect. 1 COMPCTL register is locked. Only a system reset can clear this bit. Reset type: SYSRSn

26.8.2.25 DACHVALS2 Register (Offset = 38h) [Reset = 0000h]

DACHVALS2 is shown in Figure 26-31 and described in Table 26-29.

Return to the Summary Table.

CMPSS High DAC Value Shadow Register 2

Figure 26-31. DACHVALS2 Register

15	14	13	12	11	10	9	8
RESERVED				DACVAL			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
DACVAL							
R/W-0h							

Table 26-29. DACHVALS2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DACVAL	R/W	0h	High DAC shadow register2 value. When COMPDACCTL[DACSOURCE]=0, the value of DACHVALS2 is loaded into DACHVALA when DE mode is enabled and selected DEACTIVE input is asserted. Reset type: SYSRSn

26.8.2.26 DACLVALS2 Register (Offset = 3Ah) [Reset = 0000h]

DACLVALS2 is shown in Figure 26-32 and described in Table 26-30.

Return to the Summary Table.

CMPSS Low DAC Value Shadow Register 2

Figure 26-32. DACLVALS2 Register

15	14	13	12	11	10	9	8
RESERVED				DACVAL			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
DACVAL							
R/W-0h							

Table 26-30. DACLVALS2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DACVAL	R/W	0h	Low DAC shadow register2 value. Value of DACHVALS2 is loaded into DACHVALA when DE mode is enabled and selected DEACTIVE input is asserted. Reset type: SYSRSn

26.8.2.27 COMPDACLCTL Register (Offset = 48h) [Reset = 0000h]

COMPDACLCTL is shown in Figure 26-33 and described in Table 26-31.

Return to the Summary Table.

CMPSS Low DAC Control Register

Figure 26-33. COMPDACLCTL Register

15	14	13	12	11	10	9	8
RESERVED		RAMPDIR	BLANKEN	BLANKSOURCE			
R-0h		R/W-0h	R/W-0h	R/W-0h			
7	6	5	4	3	2	1	0
RESERVED	RAMPLOADSEL	RESERVED	RAMPSOURCE				DACSOURCE
R-0h	R/W-0h	R-0h	R/W-0h				R/W-0h

Table 26-31. COMPDACLCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	RESERVED	R	0h	Reserved
13	RAMPDIR	R/W	0h	Low Ramp Generator Direction control bit. 0 Decrementing Ramp. 1 Incrementing Ramp. Reset type: SYSRSn
12	BLANKEN	R/W	0h	COMPL EPWMBLANK enable. This bit enables the EPWMBLANK signal. 0 EPWMBLANK signal is disabled. 1 EPWMBLANK signal is enabled. Reset type: SYSRSn
11-8	BLANKSOURCE	R/W	0h	COMPL EPWMBLANK source select. This bit field determines which EPWMnBLANK is passed on as the EPWMBLANK signal. Where n represents the maximum number of EPWMBLANK signals available on the device: 0 EPWM1BLANK 1 EPWM2BLANK 2 EPWM3BLANK ... n-1 EPWMnBLANK Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6	RAMPLOADSEL	R/W	0h	Ramp load select. Determines whether RAMPLSTS is updated from RAMPLREFA or RAMPLREFS when COMPSTS[COMPLSTS] is triggered. 0 RAMPLSTS is loaded from RAMPLREFA 1 RAMPLSTS is loaded from RAMPLREFS Reset type: SYSRSn
5	RESERVED	R	0h	Reserved
4-1	RAMPSOURCE	R/W	0h	Low Ramp generator source select. Determines which EPWMSYNCPER signal is used within the COMPL Where n represents the maximum number of EPWMSYNCPER signals available on the device: 0 EPWM1SYNCPER 1 EPWM2SYNCPER 2 EPWM3SYNCPER ... n-1 EPWMnSYNCPER Reset type: SYSRSn

Table 26-31. COMPDACLCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	DACSOURCE	R/W	0h	DACL source select. Determines whether DACLVALA is updated from DACLVALS or from the low ramp generator. 0 DACLVALA is updated from DACLVALS 1 DACLVALA is updated from the low ramp generator Reset type: SYSRSn

26.8.2.28 COMPDACLCTL2 Register (Offset = 4Ah) [Reset = 0000h]

COMPDACLCTL2 is shown in Figure 26-34 and described in Table 26-32.

Return to the Summary Table.

CMPSS Low DAC Control Register 2

Figure 26-34. COMPDACLCTL2 Register

15	14	13	12	11	10	9	8
RESERVED					RAMPSOURCEUSEL	RESERVED	BLANKSOURCEUSEL
R-0h					R/W-0h	R-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

Table 26-32. COMPDACLCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R	0h	Reserved
10	RAMPSOURCEUSEL	R/W	0h	0: Selects EPWM1 to 16 as RAMP source for RAMPL 1: Selects EPWM17 to 32 as RAMP source for RAMPL Reset type: SYSRSn
9	RESERVED	R	0h	Reserved
8	BLANKSOURCEUSEL	R/W	0h	0: Selects EPWM1 to 16 as BLANK source for COMPL 1: Selects EPWM17 to 32 as BLANK source for COMPL Reset type: SYSRSn
7-0	RESERVED	R	0h	Reserved

26.8.2.29 RAMPLREFA Register (Offset = 50h) [Reset = 0000h]

RAMPLREFA is shown in Figure 26-35 and described in Table 26-33.

Return to the Summary Table.

CMPSS Low Ramp Reference Active Register

Figure 26-35. RAMPLREFA Register

15	14	13	12	11	10	9	8
RAMPLREF							
R-0h							
7	6	5	4	3	2	1	0
RAMPLREF							
R-0h							

Table 26-33. RAMPLREFA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RAMPLREF	R	0h	Low Ramp reference active value. Latched value to be loaded into ramp generator RAMHPSTS. Reset type: SYSRSn

26.8.2.30 RAMPLREFS Register (Offset = 54h) [Reset = 0000h]

RAMPLREFS is shown in Figure 26-36 and described in Table 26-34.

Return to the Summary Table.

CMPSS Low Ramp Reference Shadow Register

Figure 26-36. RAMPLREFS Register

15	14	13	12	11	10	9	8
RAMPLREF							
R/W-0h							
7	6	5	4	3	2	1	0
RAMPLREF							
R/W-0h							

Table 26-34. RAMPLREFS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RAMPLREF	R/W	0h	Low Ramp reference shadow. Unlatched value to be loaded into ramp generator RAMPHSTS. Reset type: SYSRSn

26.8.2.31 RAMPLSTEPVALA Register (Offset = 58h) [Reset = 0000h]

RAMPLSTEPVALA is shown in Figure 26-37 and described in Table 26-35.

Return to the Summary Table.

CMPSS Low Ramp Step Value Active Register

Figure 26-37. RAMPLSTEPVALA Register

15	14	13	12	11	10	9	8
RAMPLSTEPVAL							
R-0h							
7	6	5	4	3	2	1	0
RAMPLSTEPVAL							
R-0h							

Table 26-35. RAMPLSTEPVALA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RAMPLSTEPVAL	R	0h	Low Ramp step value active. Latched value that will be subtracted from RAMPHSTS. Reset type: SYSRSn

26.8.2.32 RAMPLCTLA Register (Offset = 5Ah) [Reset = 0000h]

RAMPLCTLA is shown in Figure 26-38 and described in Table 26-36.

Return to the Summary Table.

CMPSS Low Ramp Control Active Register

Figure 26-38. RAMPLCTLA Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RAMPCLKDIV			
R-0h				R-0h			

Table 26-36. RAMPLCTLA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-0	RAMPCLKDIV	R	0h	Ramp Low Clock Divider Active Value RAMPCLK = SYSCLK/(RAMPCLKDIV+1) Reset type: SYSRSn

26.8.2.33 RAMPLSTEPVALS Register (Offset = 5Ch) [Reset = 0000h]

RAMPLSTEPVALS is shown in Figure 26-39 and described in Table 26-37.

Return to the Summary Table.

CMPSS Low Ramp Step Value Shadow Register

Figure 26-39. RAMPLSTEPVALS Register

15	14	13	12	11	10	9	8
RAMPLSTEPVAL							
R/W-0h							
7	6	5	4	3	2	1	0
RAMPLSTEPVAL							
R/W-0h							

Table 26-37. RAMPLSTEPVALS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RAMPLSTEPVAL	R/W	0h	Low Ramp step value shadow. Unlatched value to be loaded into RAMPHSTEPVALA. Reset type: SYSRSn

26.8.2.34 RAMPLCTLS Register (Offset = 5Eh) [Reset = 0000h]

RAMPLCTLS is shown in Figure 26-40 and described in Table 26-38.

Return to the Summary Table.

CMPSS Low Ramp Control Shadow Register

Figure 26-40. RAMPLCTLS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RAMPCLKDIV			
R-0h				R/W-0h			

Table 26-38. RAMPLCTLS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-0	RAMPCLKDIV	R/W	0h	Ramp Low Clock Divider Shadow Value This will be the unlatched value that will be loaded into the RAMPCLKDIV field of the RAMPCTLA register Reset type: SYSRSn

26.8.2.35 RAMPLSTS Register (Offset = 60h) [Reset = 0000h]

RAMPLSTS is shown in Figure 26-41 and described in Table 26-39.

Return to the Summary Table.

CMPSS Low Ramp Status Register

Figure 26-41. RAMPLSTS Register

15	14	13	12	11	10	9	8
RAMPLVALUE							
R-0h							
7	6	5	4	3	2	1	0
RAMPLVALUE							
R-0h							

Table 26-39. RAMPLSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RAMPLVALUE	R	0h	Low Ramp value. Present value of ramp generator. Reset type: SYSRSn

26.8.2.36 RAMPLDLYA Register (Offset = 68h) [Reset = 0000h]

RAMPLDLYA is shown in Figure 26-42 and described in Table 26-40.

Return to the Summary Table.

CMPSS Low Ramp Delay Active Register

Figure 26-42. RAMPLDLYA Register

15	14	13	12	11	10	9	8
RESERVED			DELAY				
R-0h			R-0h				
7	6	5	4	3	2	1	0
DELAY							
R-0h							

Table 26-40. RAMPLDLYA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12-0	DELAY	R	0h	Low Ramp delay active value. Latched value of the number of cycles to delay the start of the ramp generator stepper after a EPWMSYNCPER is received. Reset type: SYSRSn

26.8.2.37 RAMPLDLYS Register (Offset = 6Ah) [Reset = 0000h]

RAMPLDLYS is shown in Figure 26-43 and described in Table 26-41.

Return to the Summary Table.

CMPSS Low Ramp Delay Shadow Register

Figure 26-43. RAMPLDLYS Register

15	14	13	12	11	10	9	8
RESERVED			DELAY				
R-0h			R/W-0h				
7	6	5	4	3	2	1	0
DELAY							
R/W-0h							

Table 26-41. RAMPLDLYS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12-0	DELAY	R/W	0h	Low Ramp delay shadow value. Unlatched value to be loaded into RAMPHDLYA. Reset type: SYSRSn

26.8.2.38 CTRIPLFILCLKCTL2 Register (Offset = 6Eh) [Reset = 0000h]

CTRIPLFILCLKCTL2 is shown in Figure 26-44 and described in Table 26-42.

Return to the Summary Table.

CTRIPL Filter Clock Control Register 2

Figure 26-44. CTRIPLFILCLKCTL2 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CLKPRESCALEU							
R/W-0h							

Table 26-42. CTRIPLFILCLKCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	CLKPRESCALEU	R/W	0h	COMP Low filter sample clock prescale Upper Bits. The effective prescale value is (CLKPRESCALEH:CLKPRESCALE)+1 Reset type: SYSRSn

26.8.2.39 CTRIPHFILCLKCTL2 Register (Offset = 72h) [Reset = 0000h]

CTRIPHFILCLKCTL2 is shown in Figure 26-45 and described in Table 26-43.

Return to the Summary Table.

CTRIPH Filter Clock Control Register 2

Figure 26-45. CTRIPHFILCLKCTL2 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CLKPRESCALEU							
R/W-0h							

Table 26-43. CTRIPHFILCLKCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	CLKPRESCALEU	R/W	0h	COMP High filter sample clock prescale Upper Bits. The effective prescale value is (CLKPRESCALEH:CLKPRESCALE)+1 Reset type: SYSRSn

Chapter 27

▶ CONTROL PERIPHERALS

The following chapters describe the control peripherals.

Technical Reference Manual Overview

The block diagram is shown in Figure 27-1. This Technical Reference Manual is organized into five major sections:

- C29x SYSTEM RESOURCES

  These chapters describe the C29x CPU subsystem, C29x Boot ROM, device configuration, and other system peripherals.

- ANALOG PERIPHERALS

  These chapters describe the general analog subsystem configuration, Analog-to-Digital Converter (ADC), Buffered Digital-to-Analog Converter (DAC), and Comparator Subsystem (CMPSS).

- CONTROL PERIPHERALS

  These chapters describe the Enhanced Capture (eCAP), High-Resolution Capture (HRCAP), Enhanced Pulse-Width Modulator (ePWM) with High-Resolution Pulse-Width Modulator (HRPWM), Enhanced Quadrature Encoder Pulse (eQEP), and Sigma Delta Filter Module (SDFM) peripherals.

- COMMUNICATION PERIPHERALS

  These chapters describe the communication peripherals available to the C29x subsystem such as the EtherCAT, FSI, I2C, PMBUS, UART, LIN, SPI, and SENT.

- SECURITY PERIPHERALS

  This chapter describes the safety peripherals available to the C29x subsystem such as the Hardware Security Module (HSM) and Cryptographic Accelerator.

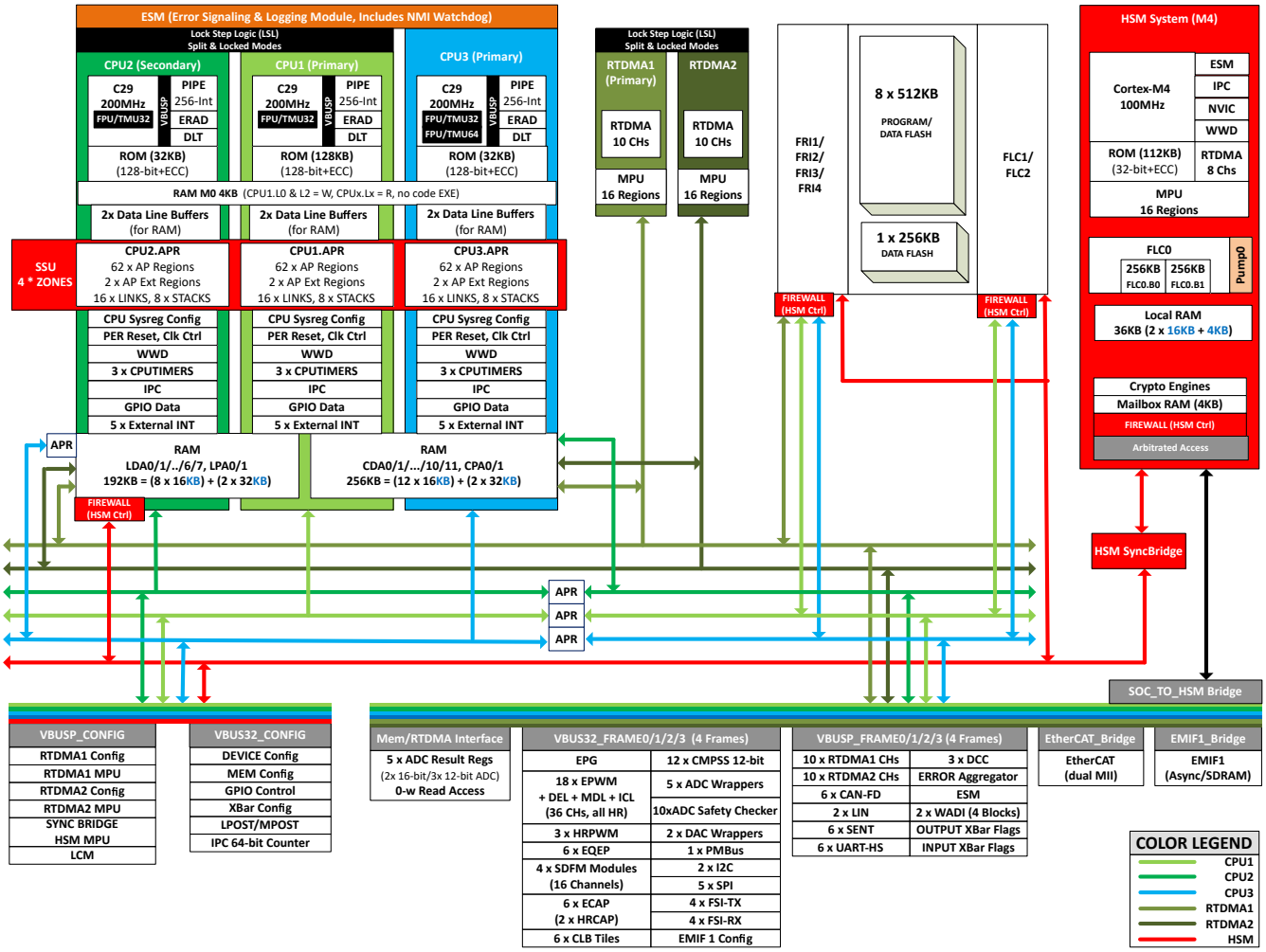


Figure 27-1. Block Diagram

Chapter 28

Enhanced Capture (eCAP)

This chapter describes the enhanced capture (eCAP) module, which is used in systems where accurate timing of external events is important.

The enhanced capture (eCAP) module is a Type 3 eCAP. See the *C2000 Real-Time Control Peripheral Reference Guide* for a list of all devices with an eCAP module of the same type, to determine the differences between the types, and for a list of device-specific differences within a type.

28.1 Introduction

28.1.1 Features

The features of the eCAP module include:

- Speed measurements of rotating machinery (for example, toothed sprockets sensed by way of Hall sensors)
- Elapsed time measurements between position sensor pulses
- Period and duty cycle measurements of pulse train signals
- Decoding current or voltage amplitude derived from duty cycle encoded current/voltage sensors

The eCAP module features described in this chapter include:

- 4-event time-stamp registers (each 32 bits)
- Edge polarity selection for up to four sequenced time-stamp capture events
- Interrupt on either of the four events
- Single-shot capture of up to four event time-stamps
- Continuous mode capture of time stamps in a four-deep circular buffer
- Absolute time-stamp capture
- Difference (Delta) mode time-stamp capture
- When not used in capture mode, the eCAP module can be configured as a single-channel PWM output

The capture functionality of the Type 1 eCAP is enhanced from the Type 0 eCAP with the following added features:

- Event filter reset bit:
  - Writing a 1 to ECCTL2[CTRFILTRESET] clears the event filter, the modulo counter, and any pending interrupts flags. Resetting the bit is useful for initialization and debug. Note that this is not applicable for signal monitoring interrupts, which do not get affected by the event filter reset bit.
- Modulo counter status bits:
  - The modulo counter (ECCTL2 [MODCNTRSTS]) indicates which capture register is loaded next. In the Type 0 eCAP, to know the current state of the modulo counter was not possible
- RTDMA trigger source:
  - eCAPxDMA was added as a RTDMA trigger. CEVT[1-4] can be configured as the source for eCAPxDMA.
- Input multiplexer:
  - ECCTL0 [INPUTSEL] selects one of 128 input signals, which are detailed in Section 28.3.
- EALLOW protection:
  - EALLOW protection was added to critical registers. To maintain software compatibility with Type-0, configure DEV_CFG_REGS.ECAPTYPE to make these registers unprotected.

The capture functionality of the Type 2 eCAP is enhanced from the Type 1 eCAP with the following added features:

- Added ECAPxSYNCINSEL register:
  - ECAPxSYNCINSEL register is added for each eCAP to select an external SYNCIN. Every eCAP can have a separate SYNCIN signal.

The capture functionality of the Type 3 eCAP is enhanced from the Type 2 eCAP with the following added features:

- Two signal monitoring units to monitor edge, pulse width, and period
  - Signal monitoring can optionally be tightly coupled with ePWM global load strobes and trip events
- Increased the number of multiplexed capture inputs from 128 to 256
- RTDMA event generation capability in PWM mode of operation
- ADC SOC generation capability, to trigger ADC conversion

28.1.2 ECAP Related Collateral

Foundational Materials

- C28x Academy - ECAP
- C29x Academy - Enhanced Capture Module (ECAP)

Getting Started Materials

- *Leveraging High Resolution Capture (HRCAP) for Single Wire Data Transfer Application Report*

28.2 Description

The eCAP module represents one complete capture channel that can be instantiated multiple times, depending on the target device. In the context of this guide, one eCAP channel has the following independent key resources:

- Capture inputs can be connected using the Input X-BAR
- 256:1 input multiplexer
- Output X-BAR is used to configure output in APWM mode
- 32-bit time base (counter)
- 4 x 32-bit time-stamp capture registers (CAP1-CAP4)
- Four-stage sequencer (modulo4 counter) that is synchronized to external events, eCAP pin rising/falling edges.
- Modulo counter status register (MODCNTRSTS) to indicate sequencer state
- Independent edge polarity (rising/falling edge) selection for all four events
- Input capture signal prescaling (from 2-62 or bypass)
- One-shot compare register (two bits) to freeze captures after 1-4 time-stamp events
- Control for continuous time-stamp captures using a four-deep circular buffer (CAP1-CAP4) scheme
- Interrupt capabilities on any of the four capture events
- Separate RTDMA trigger
- EALLOW protection to control registers
- Signal monitoring capability for edge, pulse width, and period
- RTDMA event generation capability in APWM mode
- ADC SOC event generation capability, to trigger ADC conversion

28.3 Configuring Device Pins for the eCAP

The Input X-BAR connects the device pins to the module as input. Any GPIO on the device can be configured as an input. The GPIO input qualification can be set to synchronous or asynchronous mode by setting the GPxQSELn register bits. Using synchronized inputs can help with noise immunity but affects the eCAP accuracy by ±2 cycles. The internal pull-ups can be configured in the GPyPUD register. Since the GPIO mode is used, the GPyINV register can invert the signals.

New to the Type 1 eCAP module, a 128:1 input multiplexer must also be configured (see Figure 28-3). This multiplexer can select a variety of inputs detailed in Table 28-1 by configuring ECCTL0.INPUTSEL.

Table 28-1. eCAP Input Selection

Selection of ECAP Input	ECAP1 INDEX	ECAP2 INDEX	ECAP3 INDEX	ECAP4 INDEX	ECAP5 INDEX	ECAP6 INDEX
INPUTXBAR1	0	0	0	0	0	0
INPUTXBAR2	1	1	1	1	1	1
INPUTXBAR3	2	2	2	2	2	2
INPUTXBAR4	3	3	3	3	3	3
INPUTXBAR5	4	4	4	4	4	4
INPUTXBAR6	5	5	5	5	5	5
INPUTXBAR7	6	6	6	6	6	6
INPUTXBAR8	7	7	7	7	7	7
INPUTXBAR9	8	8	8	8	8	8
INPUTXBAR10	9	9	9	9	9	9
INPUTXBAR11	10	10	10	10	10	10
INPUTXBAR12	11	11	11	11	11	11
INPUTXBAR13	12	12	12	12	12	12
INPUTXBAR14	13	13	13	13	13	13
INPUTXBAR15	14	14	14	14	14	14
INPUTXBAR16	15	15	15	15	15	15
CLB1_OUT14	16	16	Reserved	Reserved	Reserved	Reserved
CLB1_OUT15	17	17	Reserved	Reserved	Reserved	Reserved
CLB2_OUT14	Reserved	Reserved	16	16	16	Reserved
CLB2_OUT15	Reserved	Reserved	17	17	17	Reserved
CLB3_OUT14	Reserved	Reserved	Reserved	Reserved	Reserved	16
CLB3_OUT15	Reserved	Reserved	Reserved	Reserved	Reserved	17
CLB4_OUT14	Reserved	Reserved	Reserved	Reserved	Reserved	18
CLB4_OUT15	Reserved	Reserved	Reserved	Reserved	Reserved	19
CLB5_OUT14	18	18	Reserved	Reserved	Reserved	Reserved
CLB5_OUT15	19	19	Reserved	Reserved	Reserved	Reserved
CLB6_OUT14	Reserved	Reserved	18	18	18	Reserved
CLB6_OUT15	Reserved	Reserved	19	19	19	Reserved
OUTPUTXBAR1	20	20	20	20	20	20
OUTPUTXBAR2	21	21	21	21	21	21
OUTPUTXBAR3	22	22	22	22	22	22

Table 28-1. eCAP Input Selection (continued)

Selection of ECAP Input	ECAP1 INDEX	ECAP2 INDEX	ECAP3 INDEX	ECAP4 INDEX	ECAP5 INDEX	ECAP6 INDEX
OUTPUTXBAR4	23	23	23	23	23	23
OUTPUTXBAR5	24	24	24	24	24	24
OUTPUTXBAR6	25	25	25	25	25	25
OUTPUTXBAR7	26	26	26	26	26	26
OUTPUTXBAR8	27	27	27	27	27	27
ADCEEVT1	28	28	28	28	28	28
ADCEEVT2	29	29	29	29	29	29
ADCEEVT3	30	30	30	30	30	30
ADCEEVT4	31	31	31	31	31	31
ADCDEVT1	32	32	32	32	32	32
ADCDEVT2	33	33	33	33	33	33
ADCDEVT3	34	34	34	34	34	34
ADCDEVT4	35	35	35	35	35	35
ADCCEVT1	36	36	36	36	36	36
ADCCEVT2	37	37	37	37	37	37
ADCCEVT3	38	38	38	38	38	38
ADCCEVT4	39	39	39	39	39	39
ADCBEVT1	40	40	40	40	40	40
ADCBEVT2	41	41	41	41	41	41
ADCBEVT3	42	42	42	42	42	42
ADCBEVT4	43	43	43	43	43	43
ADCAEVT1	44	44	44	44	44	44
ADCAEVT2	45	45	45	45	45	45
ADCAEVT3	46	46	46	46	46	46
ADCAEVT4	47	47	47	47	47	47
FSIRXA_MEASURE	48	48	48	48	48	48
FSIRXA_MEASURE_RISE	49	49	49	49	49	49
FSIRXA_MEASURE_FALL	50	50	50	50	50	50
FSIRXB_MEASURE	51	51	51	51	51	51
FSIRXB_MEASURE_RISE	52	52	52	52	52	52
FSIRXB_MEASURE_FALL	53	53	53	53	53	53
FSIRXC_MEASURE	54	54	54	54	54	54
FSIRXC_MEASURE_RISE	55	55	55	55	55	55
FSIRXC_MEASURE_FALL	56	56	56	56	56	56
FSIRXD_MEASURE	57	57	57	57	57	57
FSIRXD_MEASURE_RISE	58	58	58	58	58	58
FSIRXD_MEASURE_FALL	59	59	59	59	59	59
SD2FLT1_COMPL	60	60	60	60	60	60

Table 28-1. eCAP Input Selection (continued)

Selection of ECAP Input	ECAP1 INDEX	ECAP2 INDEX	ECAP3 INDEX	ECAP4 INDEX	ECAP5 INDEX	ECAP6 INDEX
SD2FLT2_COMPL	61	61	61	61	61	61
SD2FLT3_COMPL	62	62	62	62	62	62
SD2FLT4_COMPL	63	63	63	63	63	63
SD1FLT1_COMPL	64	64	64	64	64	64
SD1FLT2_COMPL	65	65	65	65	65	65
SD1FLT3_COMPL	66	66	66	66	66	66
SD1FLT4_COMPL	67	67	67	67	67	67
SD2FLT1_COMPZ	68	68	68	68	68	68
SD2FLT2_COMPZ	69	69	69	69	69	69
SD2FLT3_COMPZ	70	70	70	70	70	70
SD2FLT4_COMPZ	71	71	71	71	71	71
SD1FLT1_COMPZ	72	72	72	72	72	72
SD1FLT2_COMPZ	73	73	73	73	73	73
SD1FLT3_COMPZ	74	74	74	74	74	74
SD1FLT4_COMPZ	75	75	75	75	75	75
SD2FLT1_COMPH	76	76	76	76	76	76
SD2FLT2_COMPH	77	77	77	77	77	77
SD2FLT3_COMPH	78	78	78	78	78	78
SD2FLT4_COMPH	79	79	79	79	79	79
SD1FLT1_COMPH	80	80	80	80	80	80
SD1FLT2_COMPH	81	81	81	81	81	81
SD1FLT3_COMPH	82	82	82	82	82	82
SD1FLT4_COMPH	83	83	83	83	83	83
SD2FLT1_COMPH_OR_COMPL	84	84	84	84	84	84
SD2FLT2_COMPH_OR_COMPL	85	85	85	85	85	85
SD2FLT3_COMPH_OR_COMPL	86	86	86	86	86	86
SD2FLT4_COMPH_OR_COMPL	87	87	87	87	87	87
SD1FLT1_COMPH_OR_COMPL	88	88	88	88	88	88
SD1FLT2_COMPH_OR_COMPL	89	89	89	89	89	89
SD1FLT3_COMPH_OR_COMPL	90	90	90	90	90	90
SD1FLT4_COMPH_OR_COMPL	91	91	91	91	91	91
ECAP6_DELAY_CLK	Reserved	Reserved	Reserved	Reserved	92	Reserved
ECAP5_DELAY_CLK	Reserved	Reserved	Reserved	Reserved	Reserved	93
ECAT_SYNC0	94	94	94	94	94	94
ECAT_SYNC1	95	95	95	95	95	95
CMPSS1_CTRIPL	96	96	96	96	96	96
CMPSS2_CTRIPL	97	97	97	97	97	97
CMPSS3_CTRIPL	98	98	98	98	98	98

Table 28-1. eCAP Input Selection (continued)

Selection of ECAP Input	ECAP1 INDEX	ECAP2 INDEX	ECAP3 INDEX	ECAP4 INDEX	ECAP5 INDEX	ECAP6 INDEX
CMPSS4_CTRIPL	99	99	99	99	99	99
CMPSS5_CTRIPL	100	100	100	100	100	100
CMPSS6_CTRIPL	101	101	101	101	101	101
CMPSS7_CTRIPL	102	102	102	102	102	102
CMPSS8_CTRIPL	103	103	103	103	103	103
Reserved	104-106	104-106	104-106	104-106	104-106	104-106
CMPSS1_CTRIPH	107	107	107	107	107	107
CMPSS2_CTRIPH	108	108	108	108	108	108
CMPSS3_CTRIPH	109	109	109	109	109	109
CMPSS4_CTRIPH	110	110	110	110	110	110
CMPSS5_CTRIPH	111	111	111	111	111	111
CMPSS6_CTRIPH	112	112	112	112	112	112
CMPSS7_CTRIPH	113	113	113	113	113	113
CMPSS8_CTRIPH	114	114	114	114	114	114
GPIO8	115	115	115	115	115	115
GPIO9	116	116	116	116	116	116
GPIO22	117	117	117	117	117	117
GPIO23	118	118	118	118	118	118
CMPSS1_CTRIPH_OR_CTRIPL	119	119	119	119	119	119
CMPSS2_CTRIPH_OR_CTRIPL	120	120	120	120	120	120
CMPSS3_CTRIPH_OR_CTRIPL	121	121	121	121	121	121
CMPSS4_CTRIPH_OR_CTRIPL	122	122	122	122	122	122
CMPSS5_CTRIPH_OR_CTRIPL	123	123	123	123	123	123
CMPSS6_CTRIPH_OR_CTRIPL	124	124	124	124	124	124
CMPSS7_CTRIPH_OR_CTRIPL	125	125	125	125	125	125
CMPSS8_CTRIPH_OR_CTRIPL	126	126	126	126	126	126
INPUTXBAR7	127	Reserved	Reserved	Reserved	Reserved	Reserved
INPUTXBAR8	Reserved	127	Reserved	Reserved	Reserved	Reserved
INPUTXBAR9	Reserved	Reserved	127	Reserved	Reserved	Reserved
INPUTXBAR10	Reserved	Reserved	Reserved	127	Reserved	Reserved
INPUTXBAR11	Reserved	Reserved	Reserved	Reserved	127	Reserved
INPUTXBAR12	Reserved	Reserved	Reserved	Reserved	Reserved	127
INPUTXBAR13	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
SD4FLT1_COMPL	128	128	128	128	128	128
SD4FLT2_COMPL	129	129	129	129	129	129
SD4FLT3_COMPL	130	130	130	130	130	130
SD4FLT4_COMPL	131	131	131	131	131	131
SD3FLT1_COMPL	132	132	132	132	132	132

Table 28-1. eCAP Input Selection (continued)

Selection of ECAP Input	ECAP1 INDEX	ECAP2 INDEX	ECAP3 INDEX	ECAP4 INDEX	ECAP5 INDEX	ECAP6 INDEX
SD3FLT2_COMPL	133	133	133	133	133	133
SD3FLT3_COMPL	134	134	134	134	134	134
SD3FLT4_COMPL	135	135	135	135	135	135
SD4FLT1_COMPZ	136	136	136	136	136	136
SD4FLT2_COMPZ	137	137	137	137	137	137
SD4FLT3_COMPZ	138	138	138	138	138	138
SD4FLT4_COMPZ	139	139	139	139	139	139
SD3FLT1_COMPZ	140	140	140	140	140	140
SD3FLT2_COMPZ	141	141	141	141	141	141
SD3FLT3_COMPZ	142	142	142	142	142	142
SD3FLT4_COMPZ	143	143	143	143	143	143
SD4FLT1_COMPH	144	144	144	144	144	144
SD4FLT2_COMPH	145	145	145	145	145	145
SD4FLT3_COMPH	146	146	146	146	146	146
SD4FLT4_COMPH	147	147	147	147	147	147
SD3FLT1_COMPH	148	148	148	148	148	148
SD3FLT2_COMPH	149	149	149	149	149	149
SD3FLT3_COMPH	150	150	150	150	150	150
SD3FLT4_COMPH	151	151	151	151	151	151
SD4FLT1_COMPH_OR_COMPL	152	152	152	152	152	152
SD4FLT2_COMPH_OR_COMPL	153	153	153	153	153	153
SD4FLT3_COMPH_OR_COMPL	154	154	154	154	154	154
SD4FLT4_COMPH_OR_COMPL	155	155	155	155	155	155
SD3FLT1_COMPH_OR_COMPL	156	156	156	156	156	156
SD3FLT2_COMPH_OR_COMPL	157	157	157	157	157	157
SD3FLT3_COMPH_OR_COMPL	158	158	158	158	158	158
SD3FLT4_COMPH_OR_COMPL	159	159	159	159	159	159
EPWM18_DE_ACTIVE	160	160	160	160	160	160
EPWM17_DE_ACTIVE	161	161	161	161	161	161
EPWM16_DE_ACTIVE	162	162	162	162	162	162
EPWM15_DE_ACTIVE	163	163	163	163	163	163
EPWM14_DE_ACTIVE	164	164	164	164	164	164
EPWM13_DE_ACTIVE	165	165	165	165	165	165
EPWM12_DE_ACTIVE	166	166	166	166	166	166
EPWM11_DE_ACTIVE	167	167	167	167	167	167
EPWM10_DE_ACTIVE	168	168	168	168	168	168
EPWM9_DE_ACTIVE	169	169	169	169	169	169
EPWM8_DE_ACTIVE	170	170	170	170	170	170

Table 28-1. eCAP Input Selection (continued)

Selection of ECAP Input	ECAP1 INDEX	ECAP2 INDEX	ECAP3 INDEX	ECAP4 INDEX	ECAP5 INDEX	ECAP6 INDEX
EPWM7_DE_ACTIVE	171	171	171	171	171	171
EPWM6_DE_ACTIVE	172	172	172	172	172	172
EPWM5_DE_ACTIVE	173	173	173	173	173	173
EPWM4_DE_ACTIVE	174	174	174	174	174	174
EPWM3_DE_ACTIVE	175	175	175	175	175	175
EPWM2_DE_ACTIVE	176	176	176	176	176	176
EPWM1_DE_ACTIVE	177	177	177	177	177	177
Reserved	178-249	178-249	178-249	178-249	178-249	178-249
MCANA_INT0	250	250	250	250	250	250
GPIO11	251	251	251	251	251	251
GPIO12	252	252	252	252	252	252
GPIO13	253	253	253	253	253	253
GPIO14	254	254	254	254	254	254
EPG1_DATAOUT49	255	Reserved	Reserved	Reserved	Reserved	Reserved
EPG1_DATAOUT50	Reserved	255	Reserved	Reserved	Reserved	Reserved
EPG1_DATAOUT51	Reserved	Reserved	255	Reserved	Reserved	Reserved
EPG1_DATAOUT52	Reserved	Reserved	Reserved	255	Reserved	Reserved
EPG1_DATAOUT53	Reserved	Reserved	Reserved	Reserved	255	Reserved
EPG1_DATAOUT54	Reserved	Reserved	Reserved	Reserved	Reserved	255

The Output X-BAR must be used to connect output signals to the OUTPUTXBARx output locations. The GPIO mux must then be configured to connect the OUTPUTXBARx lines to any of several IO pins with the GPIO mux. To avoid glitches on the pins, the GPyGMUX bits must be configured first (while keeping the corresponding GPyMUX bits at the default of zero), followed by writing the GPyMUX register to the desired value.

See the *General-Purpose Input/Output (GPIO)* chapter for more details on GPIO mux, GPIO settings, and XBAR configuration.

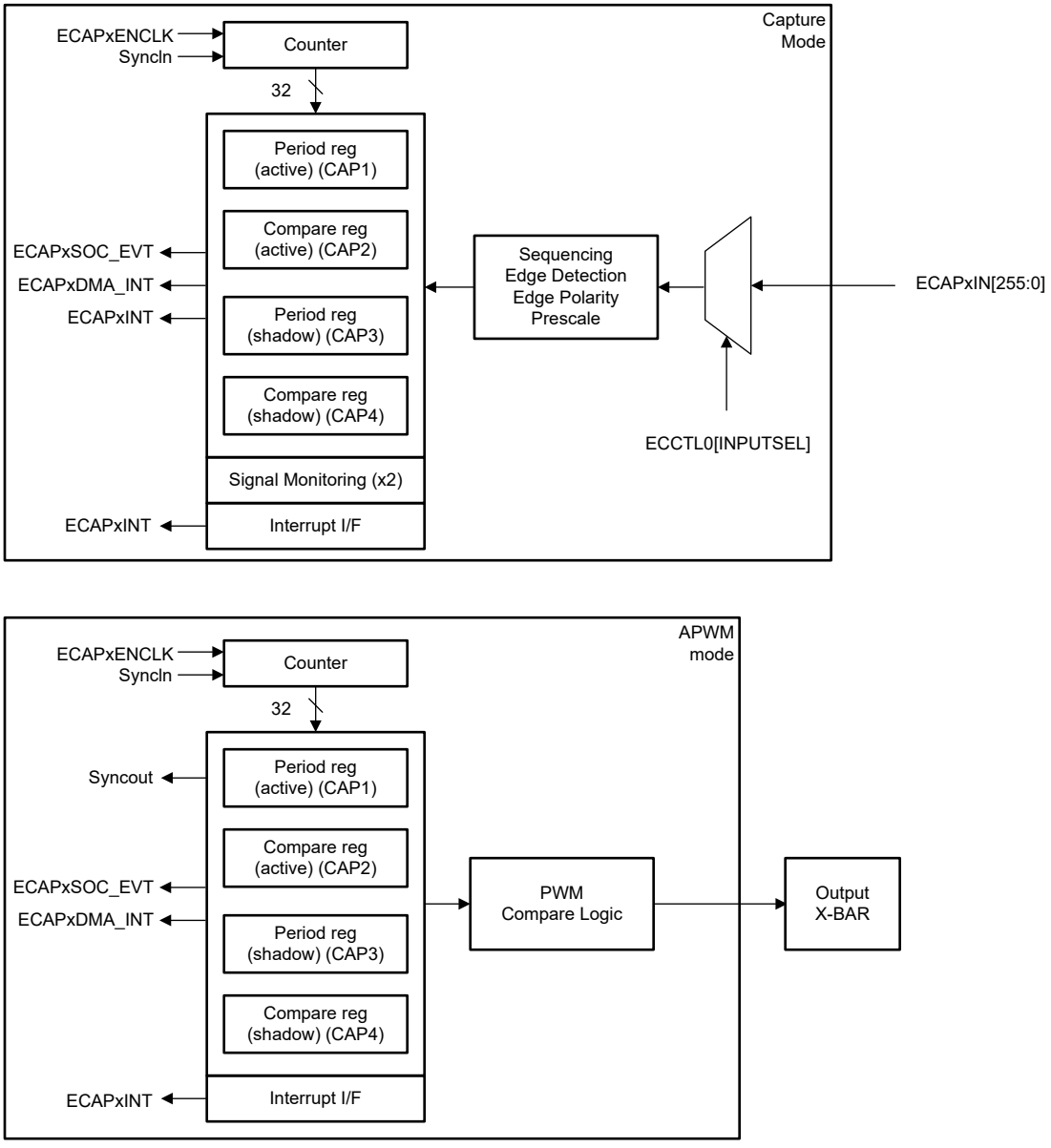
Note

ECAPxIN has to be at least 2 * SYSCLK-cycles wide to be properly captured by the eCAP module; otherwise, the input pulse can get missed from sampling by the SYSCLK.

28.4 Capture and APWM Operating Mode

Use the eCAP module resources to implement a single-channel PWM generator (with 32-bit capabilities) when the eCAP module is not being used for input captures. The counter operates in count-up mode, providing a time-base for asymmetrical pulse width modulation (PWM) waveforms. The CAP1 and CAP2 registers become the active period and compare registers, respectively, while CAP3 and CAP4 registers become the period and compare shadow registers, respectively. Figure 28-1 is a high-level view of both the capture and auxiliary pulse-width modulator (APWM) modes of operation.

Figure 28-2 further descries the output of the eCAP in APWM mode based on the CMP and PRD values.



A. A single pin is shared between CAP and APWM functions. In capture mode, the pin is an input; in APWM mode, the pin is an output.
B. In APWM mode, writing any value to CAP1/CAP2 active registers also writes the same value to the corresponding shadow registers CAP3/CAP4. This emulates immediate mode. Writing to the shadow registers CAP3/CAP4 invokes the shadow mode.

Figure 28-1. Capture and APWM Modes of Operation

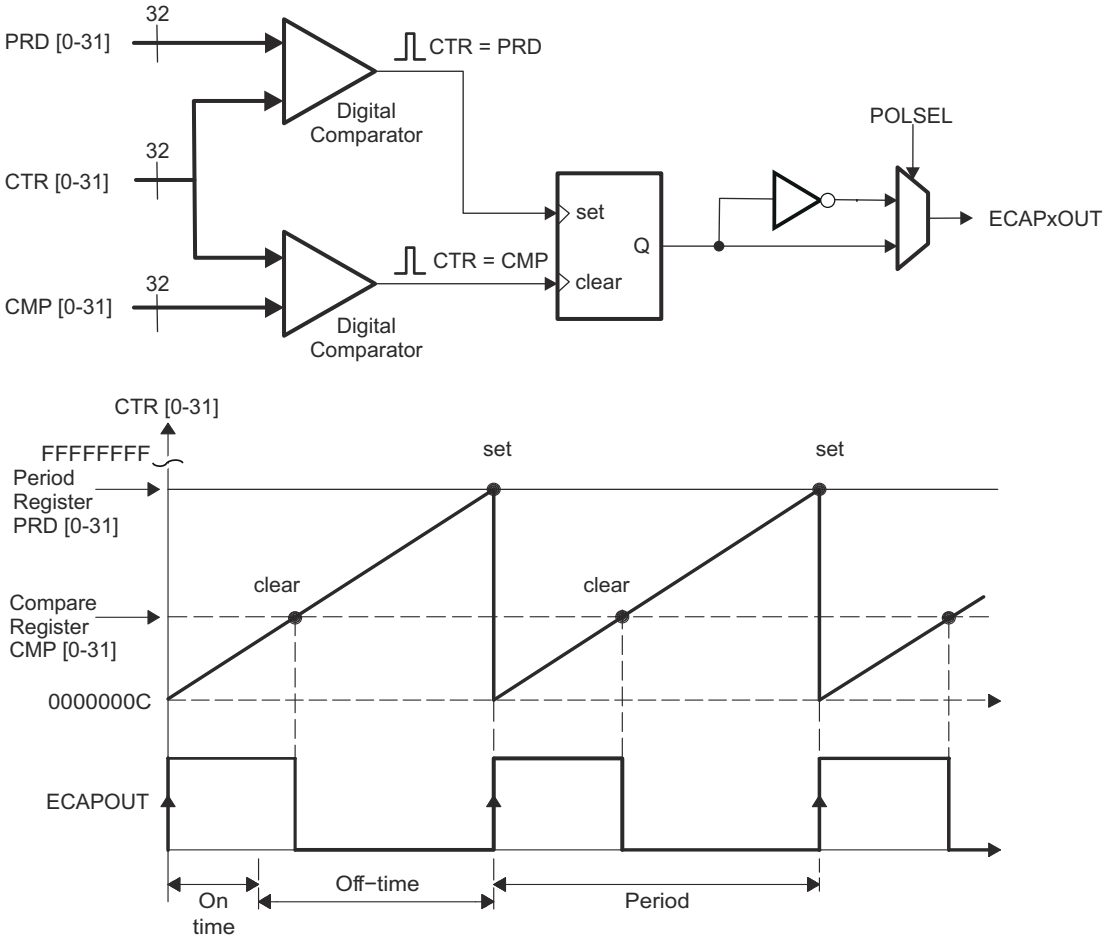
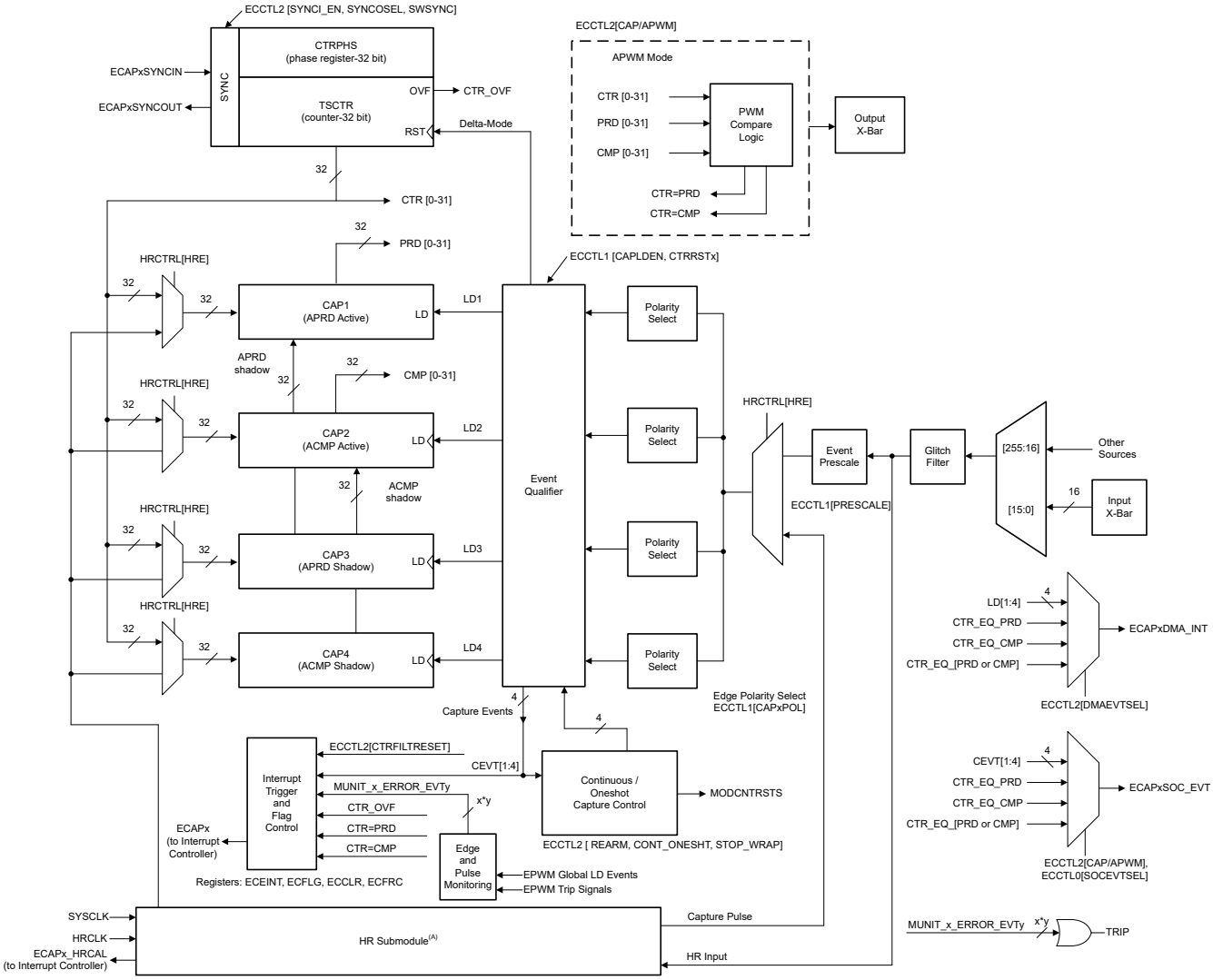


Figure 28-2. Counter Compare and PRD Effects on the eCAP Output in APWM Mode

28.5 Capture Mode Description

Figure 28-3 shows the various components that implement the capture function.

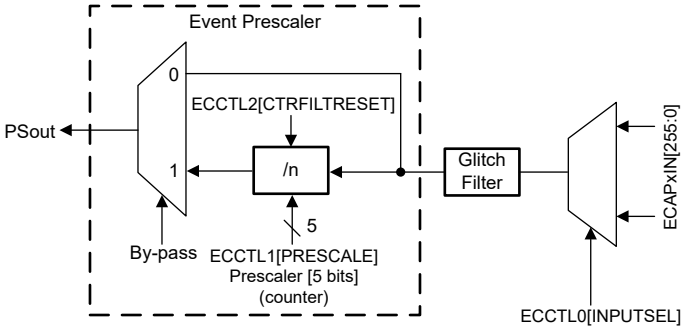


A. The HRCAP submodule is not available on all eCAP modules; in this case, the high-resolution muxes and hardware are not implemented.

Figure 28-3. eCAP Block Diagram

28.5.1 Event Prescaler

An input capture signal (pulse train) can be prescaled by N = 2-62 (in multiples of 2) or can bypass the prescaler. This is useful when very high frequency signals are used as inputs. Figure 28-4 shows a functional diagram and Figure 28-5 shows the operation of the prescale function. The event prescaler can be reset by setting the ECCTL2.CTRFILTRESET register bit.



A. When a prescale value of 1 is chosen (ECCTL1[13:9] = 0,0,0,0,0), the input capture signal bypasses the prescale logic completely.
B. The first Rise edge after Prescale configuration change is not passed to Capture logic, prescaler value takes into effect on the second rising edge after the configuration.

Figure 28-4. Event Prescale Control

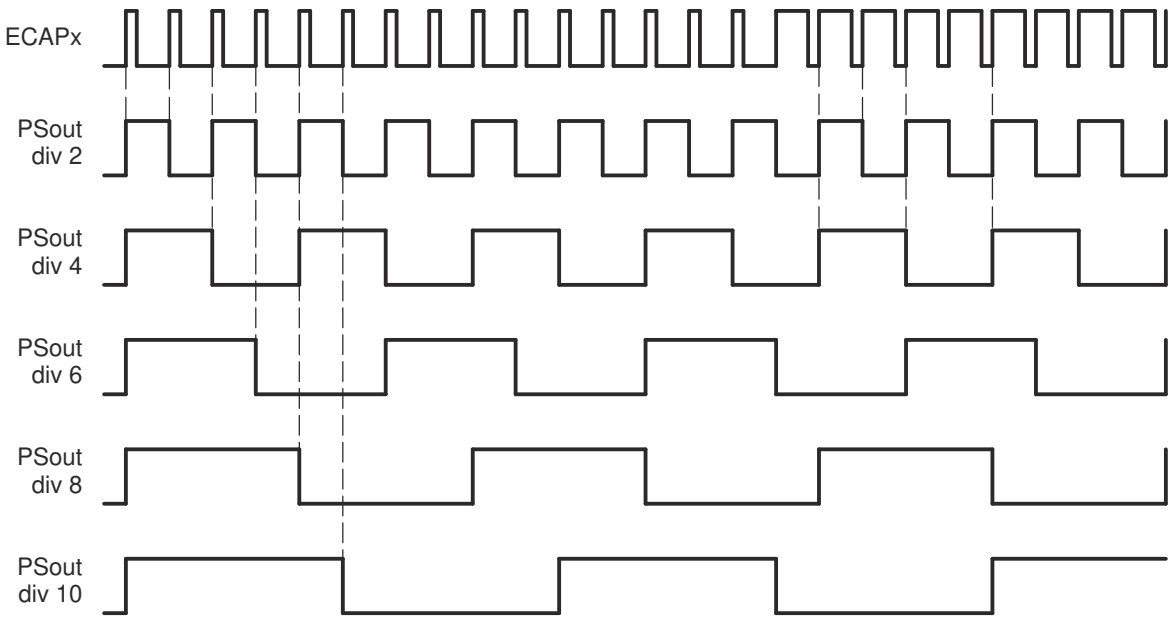


Figure 28-5. Prescale Function Waveforms

28.5.2 Glitch Filter

A glitch filter is included to reduce internal and external noise glitches on the source signal being measured by the eCAP.

The glitch filter can be used to filter out glitches of a specified time period in terms of SYSCLK cycles. The supported range is from 1 to 15 cycles. By default, the glitch filter is disabled (ECCCTL0[QUALPRD] = 0) to maintain compatibility.

Note

The glitch filter can be disabled when using HRCAP as the filter changes the pulse width of the signal. Also note that when enabled, the glitch filter delays the signal by QUALPRD+1, which must be accounted for in applications utilizing the feature.

28.5.3 Edge Polarity Select and Qualifier

Functionality and features include:

- Four independent edge polarity (rising edge/falling edge) selection muxes are used, one for each capture event.
- Each edge (up to 4) is event qualified by the Modulo4 sequencer.
- The edge event is gated to the respective CAPx register by the Mod4 counter. The CAPx register is loaded on the falling edge.

28.5.4 Continuous/One-Shot Control

Operation of eCAP in Continuous/One-Shot mode:

- The Mod4 (2-bit) counter is incremented using edge qualified events (CEVT1-CEVT4).
- The Mod4 counter continues counting (0->1->2->3->0) and wraps around unless stopped.
- During one-shot operation, a 2-bit stop register (STOP_WRAP) is used to compare the Mod4 counter output, and when equal, stops the Mod4 counter and inhibits further loads of the CAP1-CAP4 registers. In this mode, if TSCCTR counter is configured to reset on capture event (CEVTx) by configuring ECCTL1.CTRRSTx bit, the operation still keeps resetting the TSCCTR counter on capture event (CEVTx) after the STOP_WRAP value is reached and re-arm (REARM) has not occurred.

The continuous/one-shot block controls the start, stop and reset (zero) functions of the Mod4 counter, using a mono-shot type of action that can be triggered by the stop-value comparator and re-armed using software control.

Once armed, the eCAP module waits for 1-4 (defined by stop-value) capture events before freezing both the Mod4 counter and contents of CAP1-4 registers (time stamps).

Re-arming prepares the eCAP module for another capture sequence. Also, re-arming clears (to zero) the Mod4 counter and permits loading of CAP1-4 registers again, providing the CAPLDEN bit is set.

In continuous mode, the Mod4 counter continues to run (0->1->2->3->0, the one-shot action is ignored, and capture values continue to be written to CAP1-4 in a circular buffer sequence.

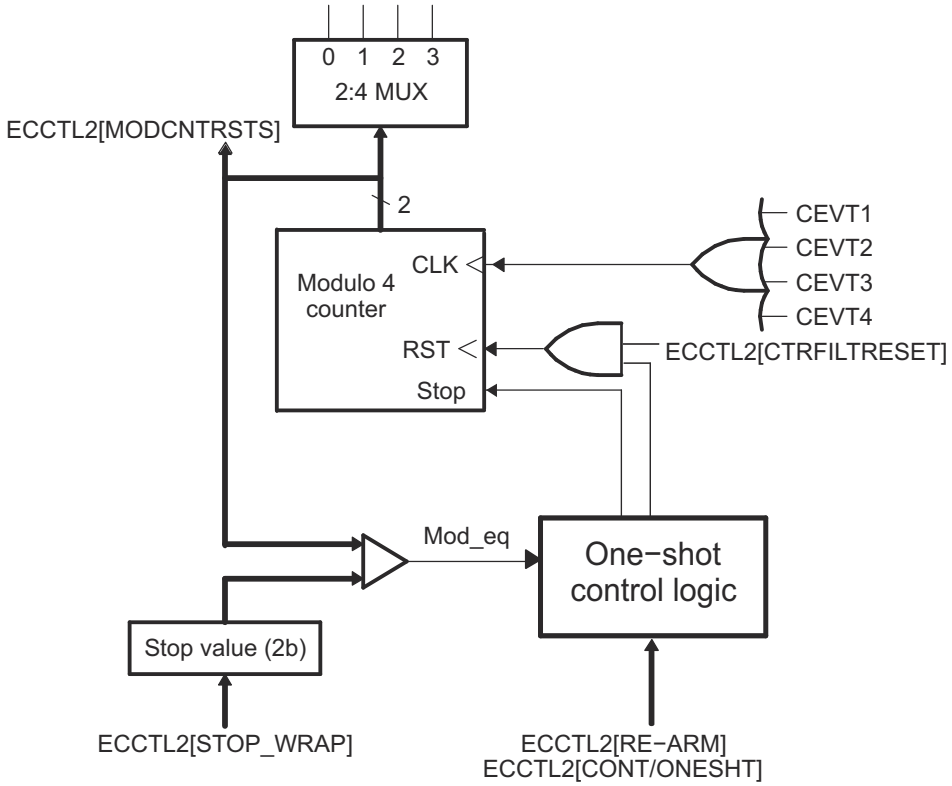


Figure 28-6. Details of the Continuous/One-shot Block

28.5.5 32-Bit Counter and Phase Control

This counter provides the time-base for event captures, and is clocked using the system clock.

A phase register is provided to achieve synchronization with other counters using a hardware and software forced sync. This is useful in APWM mode when a phase offset between modules is needed.

On any of the four event loads, an option to reset the 32-bit counter is given. This is useful for time difference capture. The 32-bit counter value is captured first, then the counter value is reset to 0 by any of the LD1-LD4 signals.

28.5.6 CAP1-CAP4 Registers

These 32-bit registers are supplied by the 32-bit counter timer bus, CTR[0-31], and are loaded (capture a time-stamp) when the respective LD inputs are strobed.

Control bit CAPLDEN can inhibit loading of the capture registers. During one-shot operation, this bit is cleared (loading is inhibited) automatically when a stop condition occurs, StopValue = Mod4.

CAP1 and CAP2 registers become the active period and compare registers, respectively, in APWM mode.

CAP3 and CAP4 registers become the respective shadow registers (APRD and ACMP) for CAP1 and CAP2 during APWM operation.

28.5.7 eCAP Synchronization

eCAP modules can be synchronized with each other by selecting a common SYNCIN source. SYNCIN source for eCAP can be either software sync-in or external sync-in. The external sync-in signal can come from eCAP, X-Bar or EPWM. The SWSYNC of the eCAP module is logical ORed with the SYNC signal as shown in Figure 28-7. The SYNC signal is defined by the selection of ECAPxSYNCINSEL[SEL] as shown in Figure 28-8.

Note

ECAPxSYNCOUT going to the ECAPSYNCIN multiplexer is disabled. For example, ECAP1SYNCOUT cannot be a sync in to ECAP1, but ECAP1SYNCOUT can be a sync in to ECAP2, ECAP3, and so on.

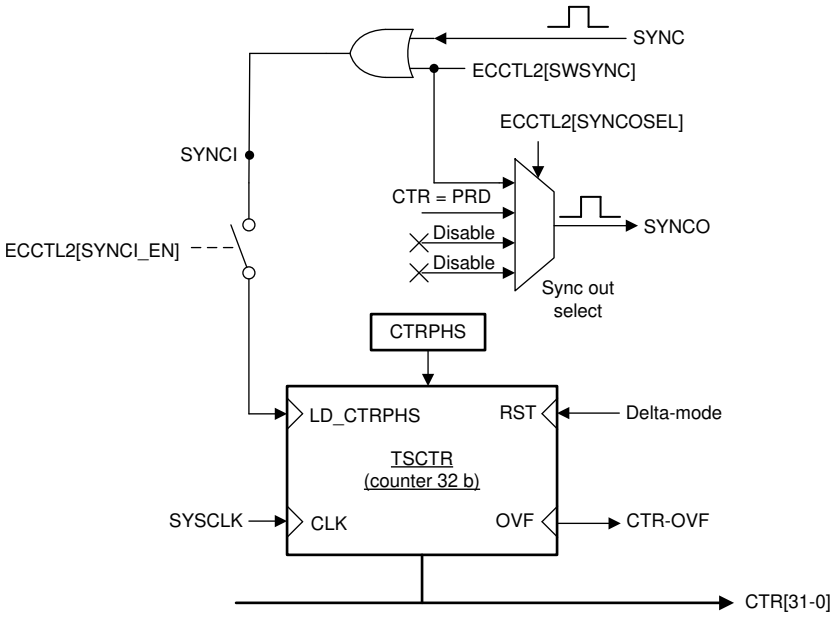


Figure 28-7. Details of the Counter and Synchronization Block

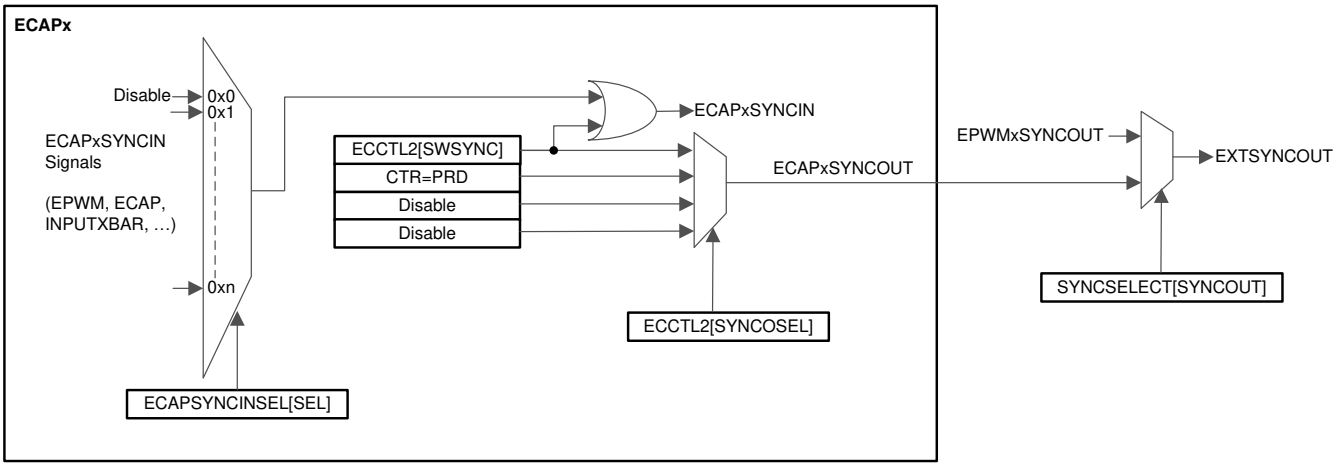


Figure 28-8. eCAP Synchronization Scheme

28.5.7.1 Example 1 - Using SWSYNC with ECAP Module

Implement the following steps to use SWSYNC with ECAP1 and ECAP2.

- Configure ECAP[1..2].ECAPSYNCINSEL.SEL = 0x0 to disable external SYNCIN coming to eCAP1.
- Configure ECAP[1..2].ECCTL2.SWSYNC = 0x1, to force Software Synchronization of the TSCTR counter.

To use SWSYNC with other eCAP modules, make sure that the previous eCAP chain is not generating a SYNCOUT signal that interferes with the software synchronization.

28.5.8 Interrupt Control

Operation and features of the eCAP interrupt control include (see Figure 28-9):

- An interrupt can be generated on capture events (CEVT1-CEVT4, CTROVF) or APWM events (CTR = PRD, CTR = CMP).
- An interrupt can be generated on signal monitoring errors (MUNIT_1_ERROR_EVT1, MUNIT_1_ERROR_EVT1, MUNIT_2_ERROR_EVT1, MUNIT_2_ERROR_EVT2)
- A counter overflow event (FFFFFFFF->00000000) is also provided as an interrupt source (CTROVF).
- The capture events are edge and sequencer-qualified (ordered in time) by the polarity select and Mod4 gating, respectively.
- One of these events can be selected as the interrupt source (from the eCAPx module) going to the Interrupt Controller.
- Seven interrupt events (CEVT1, CEVT2, CEVT3, CEVT4, CNTOVF, CTR=PRD, CTR=CMP) can be generated.
- An additional four interrupt events (MUNIT_1_ERROR_EVT1, MUNIT_1_ERROR_EVT1, MUNIT_2_ERROR_EVT1, MUNIT_2_ERROR_EVT2) can be generated from the signal monitoring unit.
- The interrupt enable register (ECEINT) is used to enable/disable individual interrupt event sources. The interrupt flag register (ECFLG) indicates if any interrupt event has been latched and contains the global interrupt flag bit (INT). An interrupt pulse is generated to the PIPE only if any of the interrupt events are enabled, the flag bit is 1, and the INT flag bit is 0. The interrupt service routine must clear the global interrupt flag bit and the serviced event using the interrupt clear register (ECCLR) before any other interrupt pulses are generated. All interrupt flags are cleared upon an event filter reset by writing a 1 to ECCTL2[CLRFILTRESET]. To force an interrupt event, use the interrupt force register (ECFRC). This is useful for test purposes.

Note

The CEVT1, CEVT2, CEVT3, CEVT4 flags are only active in capture mode (ECCTL2[CAP/APWM == 0]). The CTR=PRD, CTR=CMP flags are only valid in APWM mode (ECCTL2[CAP/APWM == 1]). CNTOVF flag is valid in both modes.

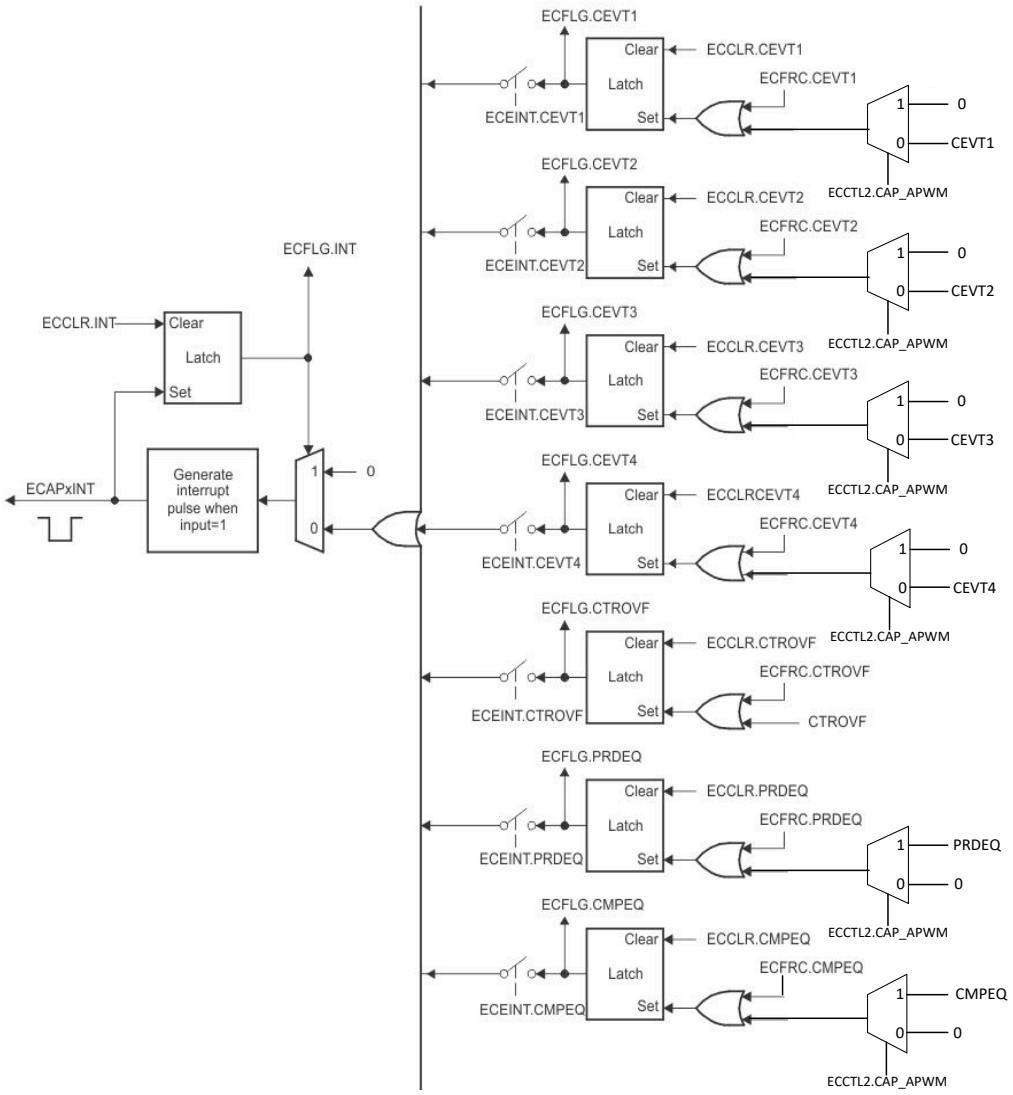


Figure 28-9. Interrupts in eCAP Module

28.5.9 RTDMA Interrupt

On Type 0 eCAP modules, the CPU was required to begin data transfers using RTDMA. New to the Type 1 eCAP, a separate RTDMA Trigger (ECAP_DMA_INT) enables continuous transfer of capture data from eCAP registers to on-chip memory using RTDMA. Any one of the four available interrupt events (LD1, LD2, LD3, and LD4) can be selected as the trigger source for ECAP_DMA_INT using ECCTL2 [DMAEVTSEL].

New to the Type 3 eCAP is the ability to trigger RTDMA events in APWM mode. Any one of three available events (period match, compare match, or both) can be selected as the trigger source for ECAP_DMA_INT using ECCTL2 [DMAEVTSEL].

28.5.10 ADC SOC Event

Type 3 introduces the capability to generate ADC SOC events in capture mode and in APWM mode of operation. The ability to start ADC conversions allows for increased APWM functionality, as well as the ability to synchronize capture events with ADC samples.

In capture mode, one of the four available interrupt events (CEVT1, CEVT2, CEVT3, and CEVT4) can be selected as ECAP_SOC_EVT using ECCCTL0[SOCEVTSEL].

In APWM mode, any one of three available events (period match, compare match, or both) can be selected as ECAP_SOC_EVT using ECCCTL0[SOCEVTSEL].

28.5.11 Shadow Load and Lockout Control

In capture mode, this logic inhibits (locks out) any shadow loading of CAP1 or CAP2 from APRD and ACMP registers, respectively.

In APWM mode, shadow loading is active and two choices are permitted:

- Immediate - APRD or ACMP are transferred to CAP1 or CAP2 immediately upon writing a new value.
- On period equal, CTR[31:0] = PRD[31:0].

28.5.12 APWM Mode Operation

Main operating highlights of the APWM section:

- The time-stamp counter bus is made available for comparison by way of 2 digital (32-bit) comparators.
- When CAP1/2 registers are not used in capture mode, the contents can be used as Period and Compare values in APWM mode.
- Double buffering is achieved using shadow registers APRD and ACMP (CAP3/4). The shadow register contents are transferred over to CAP1/2 registers, either immediately upon a write, or on a CTR = PRD trigger.
- In APWM mode, writing to CAP1/CAP2 active registers also writes the same value to the corresponding shadow registers CAP3/CAP4. This emulates immediate mode. Writing to the shadow registers CAP3/CAP4 invokes the shadow mode.
- During initialization, write to the active registers for both period and compare. This automatically copies the initial values into the shadow values. For subsequent compare updates during run-time, use the shadow registers.

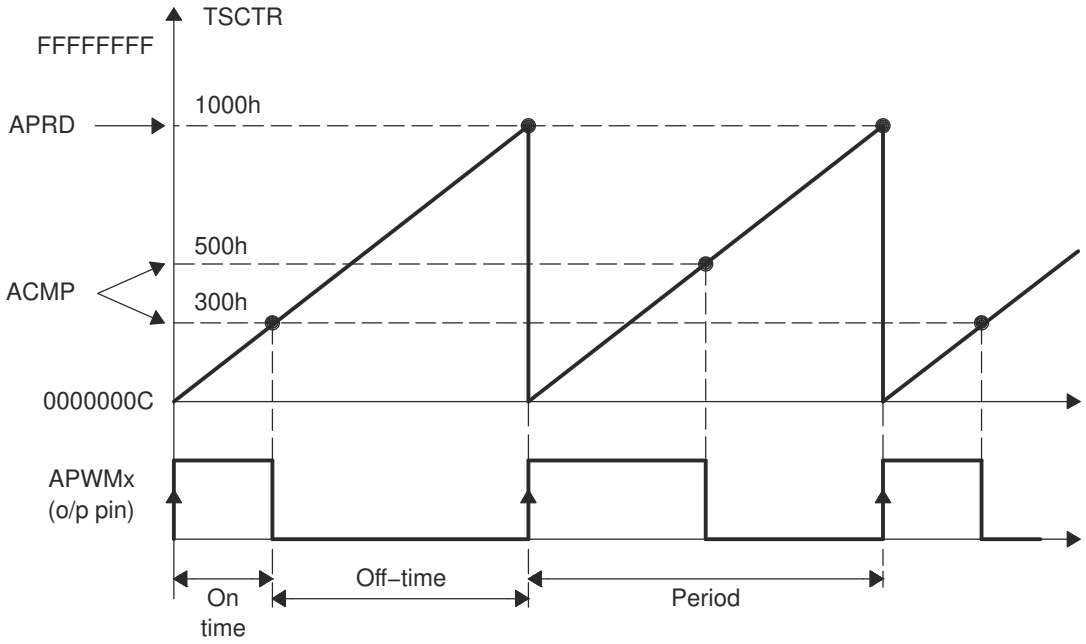


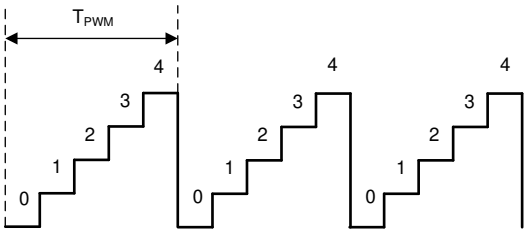
Figure 28-10. PWM Waveform Details Of APWM Mode Operation

The behavior of APWM active high mode (APWMPOL == 0) is as follows:

```
CMP = 0x00000000, output low for duration of period (0% duty)

CMP = 0x00000001, output high 1 cycle

CMP = 0x00000002, output high 2 cycles

CMP = PERIOD, output high except for 1 cycle (<100% duty)

CMP = PERIOD+1, output high for complete period (100% duty)

CMP > PERIOD+1, output high for complete period
```

The behavior of APWM active low mode (APWMPOL == 1) is as follows:

```
CMP = 0x00000000, output high for duration of period (0% duty)

CMP = 0x00000001, output low 1 cycle

CMP = 0x00000002, output low 2 cycles

CMP = PERIOD, output low except for 1 cycle (<100% duty)

CMP = PERIOD+1, output low for complete period (100% duty)

CMP > PERIOD+1, output low for complete period
```



$$T_{PWM} = (CAP1 + 1) \times T_{TSCTR}$$

$$F_{PWM} = \frac{1}{T_{PWM}}$$

Figure 28-11. Time-Base Frequency and Period Calculation

28.5.13 Signal Monitoring Unit

The signal monitoring unit can be used for edge, pulse width, and period monitoring of ECAP input signals. This allows for detection that is useful for many applications. For example, EPWM pulse width boundary monitoring can be accomplished for safety applications.

The high-level features of the signal monitoring unit include:

- Measure pulse width (high or low) and check if the pulse width is in expected range
- Measure period (rise-to-rise or fall-to-fall) and check if the period is in expected range
- Monitor signal edge (rise or fall) and check if the signal edge occurs in a user-programmed time window

28.5.13.1 Pulse Width and Period Monitoring

The signal monitoring unit has the ability to measure pulse width (either low or high) or period (rise-to-rise edge or fall-to-fall edge) and automatically generate an error when the pulse width is outside of a programmable expected range.

The expected pulse width range is programmable using the following configuration registers (or their respective shadow registers):

- MUNIT_#_MIN programs the minimum pulse width capture value
- MUNIT_#_MAX programs the maximum pulse width capture value

Any pulse width outside of these programmed bounds triggers one of two error events:

- MUNIT_#_ERROR_EVT1 generated when measured pulse width is less than MUNIT_#_MIN
- MUNIT_#_ERROR_EVT2 generated when measured pulse width is greater than MUNIT_#_MAX

The following diagram provides an example in which the measured pulse width exceeds the MAX value, generating an ERROR_EVT2 event.

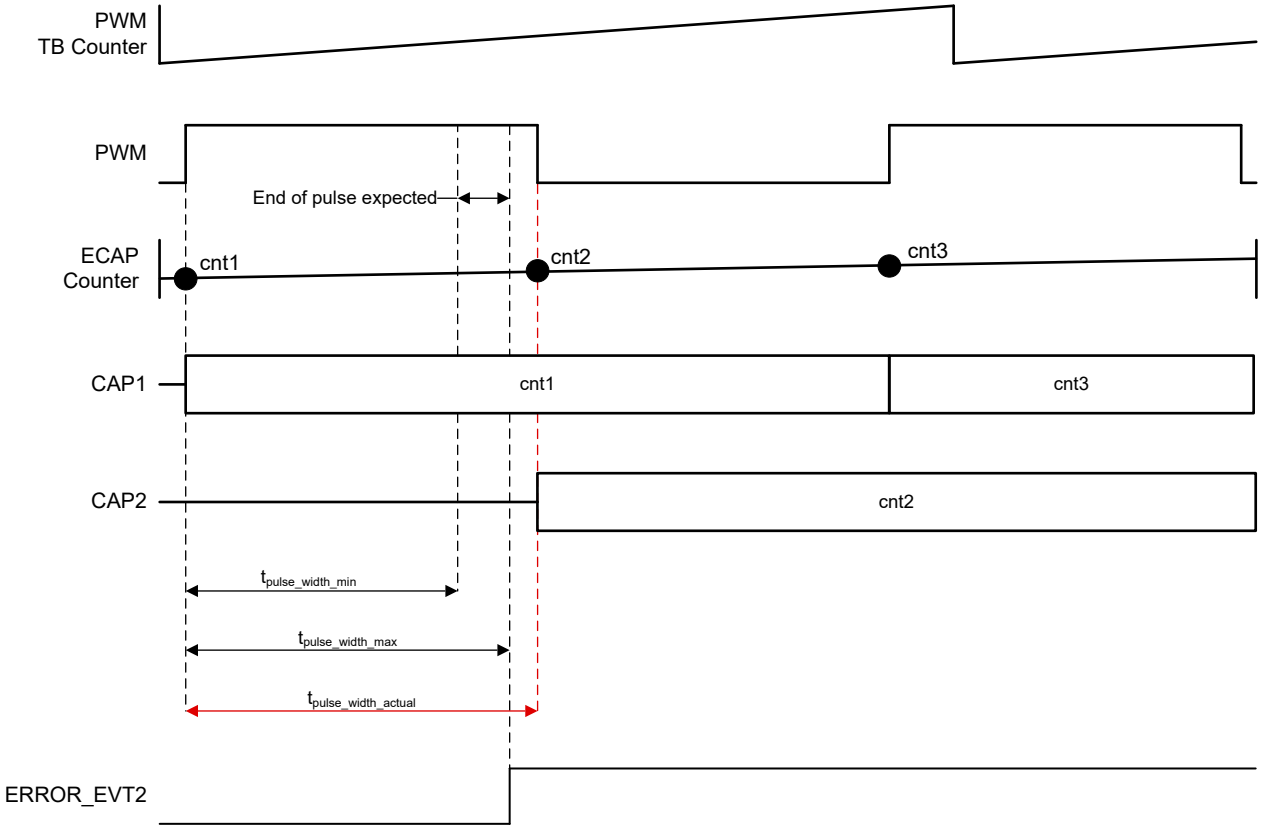


Figure 28-12. ECAP Signal Monitoring Unit Pulse Width Error Example

Configuration Requirements

To enable this mode, the following settings must be configured:

- Absolute mode must be set for the ECAP counter, so that the counter is free running and does not get reset on any capture events
- Continuous mode must be enabled (one-shot mode can be used, but is not recommended given the short duration)
- Sync feature for the counter must be disabled (ECCTL2.SYNCI_EN = 0)
- A minimum of two captures must be enabled (ECCTL2.STOP_WRAP >= 1, and at least CAP1 and CAP2 enabled)

- Capture Edge (ECCTL1.CAPxPOL) of used capture modules (any of CAP1 to CAP4) must be configured to capture two edges of interest
  - High pulse: one rising edge and one falling edge
  - Low pulse: one rising edge and one falling edge
  - Period rise-to-rise: two rising edges
  - Period fall-to-fall: two falling edges

Note

If a pulse width is greater than the MAX value, a second edge can arrive late or never even occur. Because of this, the DISABLE_EARLY_MAX_ERR field in the MUNIT_#_CTL register can be used to choose when a MAX error occurs. By setting the bit to 0, an error is generated as soon as the pulse width is greater than the specified maximum value. By setting the bit to 1, an error is generated when the second event has occurred.

28.5.13.2 Edge Monitoring

The signal monitoring unit has the ability to monitor and check if a rise or fall edge occurs within a specified time window and automatically generate an error when an edge occurs outside of this window.

The time window of an expected edge event can be programmed using the following configuration registers (or their respective shadow registers):

- MUNIT_#_MIN programs the minimum pulse width capture value
- MUNIT_#_MAX programs the maximum pulse with capture value

Any edge that occurs outside of these programmed bounds triggers the following error event:

- MUNIT_#_ERROR_EVT1 generated when edge occurs outside the bounds of MUNIT_#_MIN and MUNIT_#_MAX.

Additionally, ERROR_EVT2 is generated if either MIN or MAX did not occur between two sync events.

The following diagram provides an example in which a rising edge does not occur during the expected window, generating an ERROR_EVT1 event.

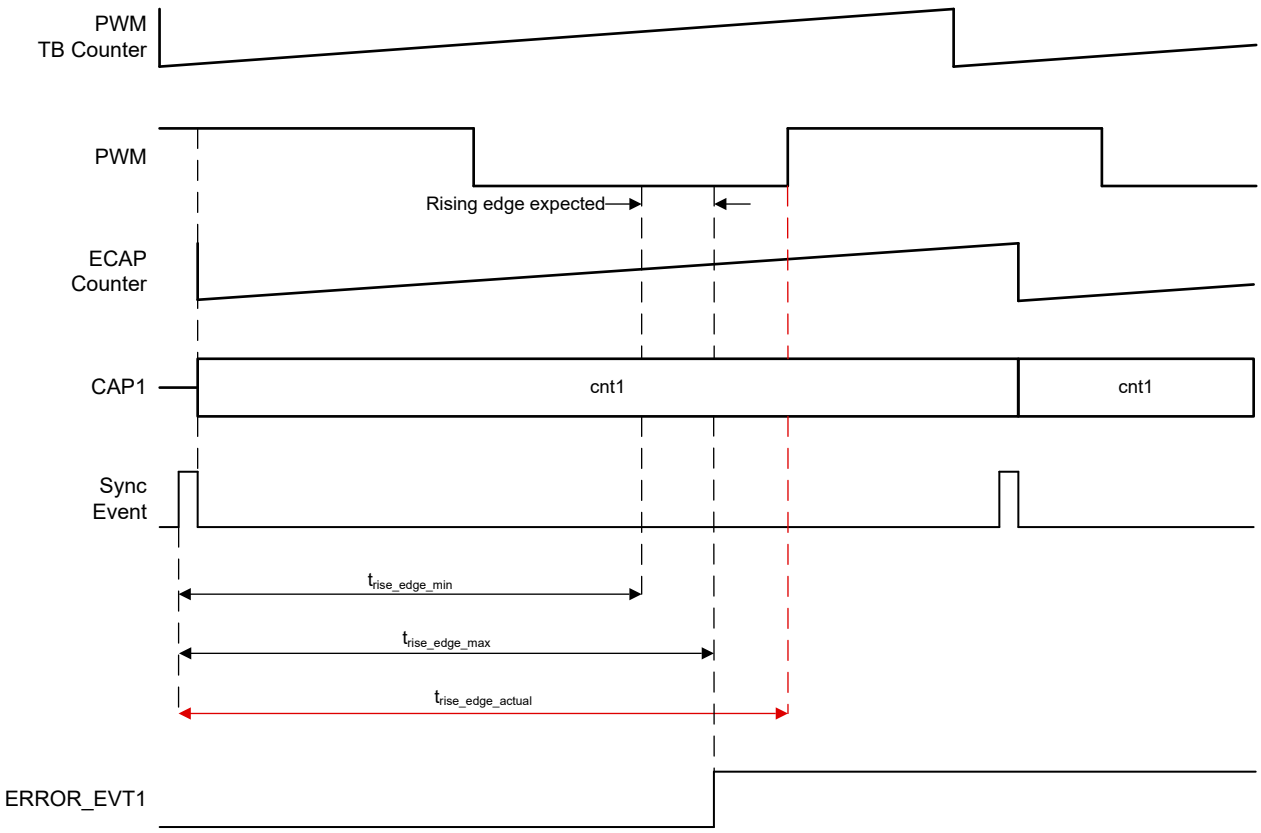


Figure 28-13. ECAP Signal Monitoring Unit Edge Error Example

Configuration Requirements

To enable this mode, the following settings must be configured:

- The ECAP counter must be synced with an EPWM module
- Absolute mode must be set for the ECAP counter, so that the counter is free running and does not get reset on any capture events
- Continuous mode can be enabled (one-shot mode can be used, but is not recommended given the modes short duration)
- A minimum of one capture can be enabled (ECCTL2.STOP_WRAP >= 0, and at least CAP1 enabled)
- Capture Edge (ECCTL1.CAPxPOL) of used capture modules (any of CAP1 to CAP4) must be configured to capture an edge of interest
- The time window defined using MIN and MAX can not cross the sync boundary

Note

The following are important considerations when configuring the edge monitoring feature:

- If the EPWM counter or ECAP counter are loaded with a non-zero phase value, the MIN and MAX values must be adjusted accordingly in SW. This also applies when the glitch filter is enabled, as the glitch filter delays the signal by QUALPRD+1
- The edge monitoring logic restarts on a sync event. This is to avoid any deadlock in case MIN, MAX, or both events do not occur between two sync events. ERROR_EVT2 is generated, if MIN or MAX match did not occur between two sync events
- The time window defined using MIN and MAX can not cross the sync boundary

28.6 Application of the eCAP Module

The following sections provide applications examples to show how to operate the eCAP module.

28.6.1 Example 1 - Absolute Time-Stamp Operation Rising-Edge Trigger

Figure 28-14 shows an example of continuous capture operation (Mod4 counter wraps around). In this figure, TSCTR counts-up without resetting and capture events are qualified on the rising edge only, this gives period (and frequency) information.

On an event, the TSCTR contents (time-stamp) is first captured, then Mod4 counter is incremented to the next state. When the TSCTR reaches FFFFFFFF (maximum value), the Mod4 counter wraps around to 00000000 (not shown in Figure 28-14), if this occurs, the CTROVF (counter overflow) flag is set, and an interrupt (if enabled) occurs. Captured Time-stamps are valid at the point indicated by the diagram (after the fourth event); hence, event CEVT4 can conveniently be used to trigger an interrupt and the CPU can read data from the CAPx registers.

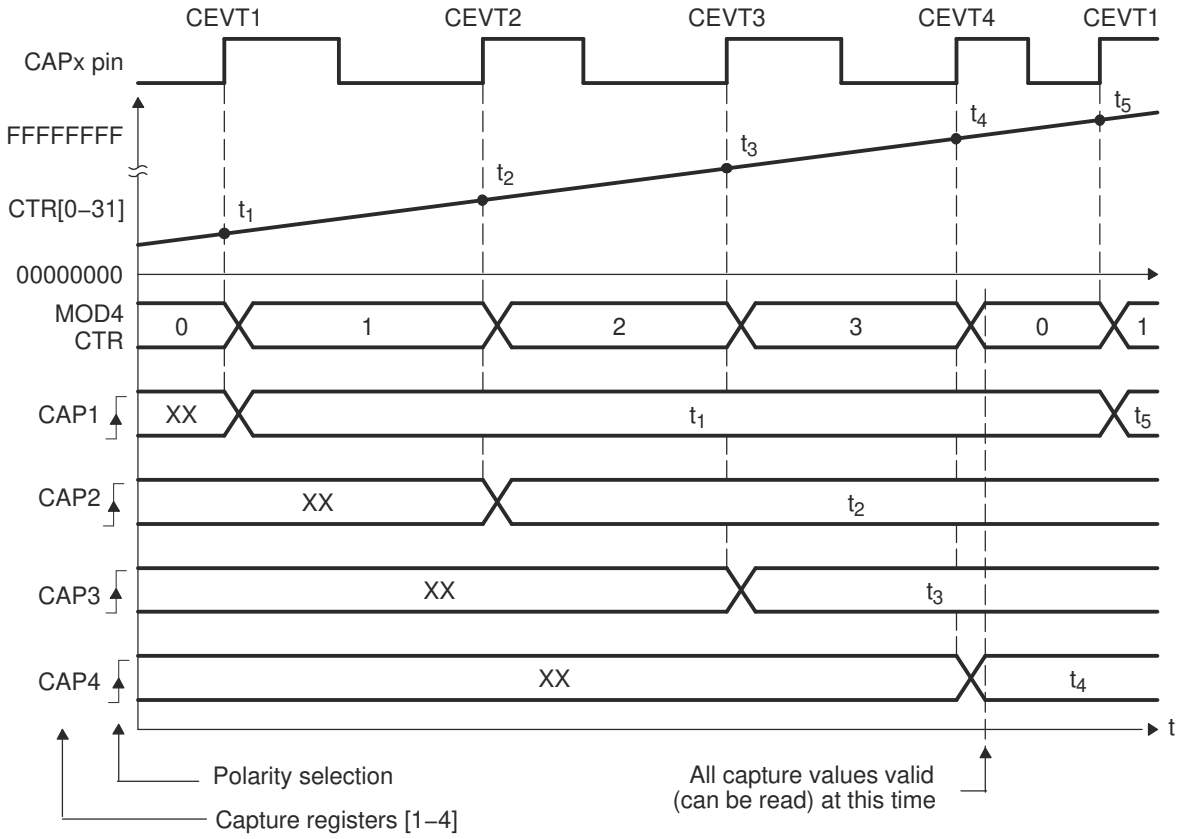


Figure 28-14. Capture Sequence for Absolute Time-stamp and Rising-Edge Detect

28.6.2 Example 2 - Absolute Time-Stamp Operation Rising- and Falling-Edge Trigger

In Figure 28-15, the eCAP operating mode is almost the same as in the previous section except capture events are qualified as either rising or falling edge, this now gives both period and duty cycle information, that is: Period1 = $t_3 - t_1$, Period2 = $t_5 - t_3$, …and so on. Duty Cycle1 (on-time %) = $(t_2 - t_1)$ / Period1 x 100%, and so on. Duty Cycle1 (off-time %) = $(t_3 - t_2)$ / Period1 x 100%, and so on.

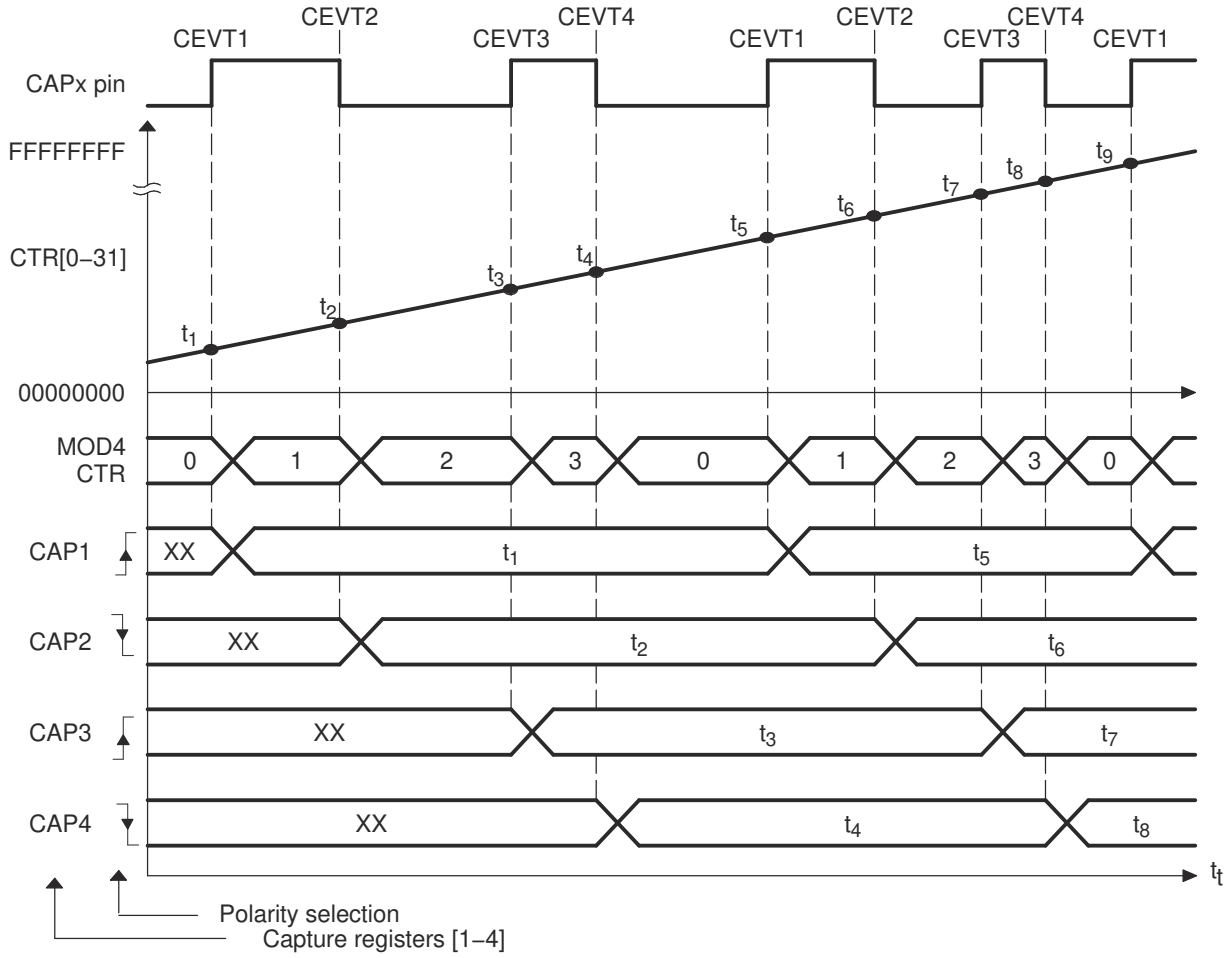


Figure 28-15. Capture Sequence for Absolute Time-stamp with Rising- and Falling-Edge Detect

28.6.3 Example 3 - Time Difference (Delta) Operation Rising-Edge Trigger

Figure 28-16 shows how the eCAP module can be used to collect delta timing data from pulse train waveforms. Here Continuous Capture mode (TSCTR counts-up without resetting, and Mod4 counter wraps around) is used. In Delta-time mode, TSCTR is reset back to zero on every valid event. Here capture events are qualified as rising edge only. On an event, TSCTR contents (Time-Stamp) is captured first, and then TSCTR is reset to zero. The Mod4 counter then increments to the next state. If TSCTR reaches FFFFFFFF (maximum value), before the next event, the Mod4 counter wraps around to 00000000 and continues, a CNTOVF (counter overflow) flag is set, and an interrupt (if enabled) occurs. The advantage of Delta-time mode is that the CAPx contents directly give timing data without the need for CPU calculations, that is, Period1 = T_1, Period2 = T_2, and so on. As shown in Figure 28-16, the CEVT1 event is a good trigger point to read the timing data, T_1, T_2, T_3, T_4 are all valid here.

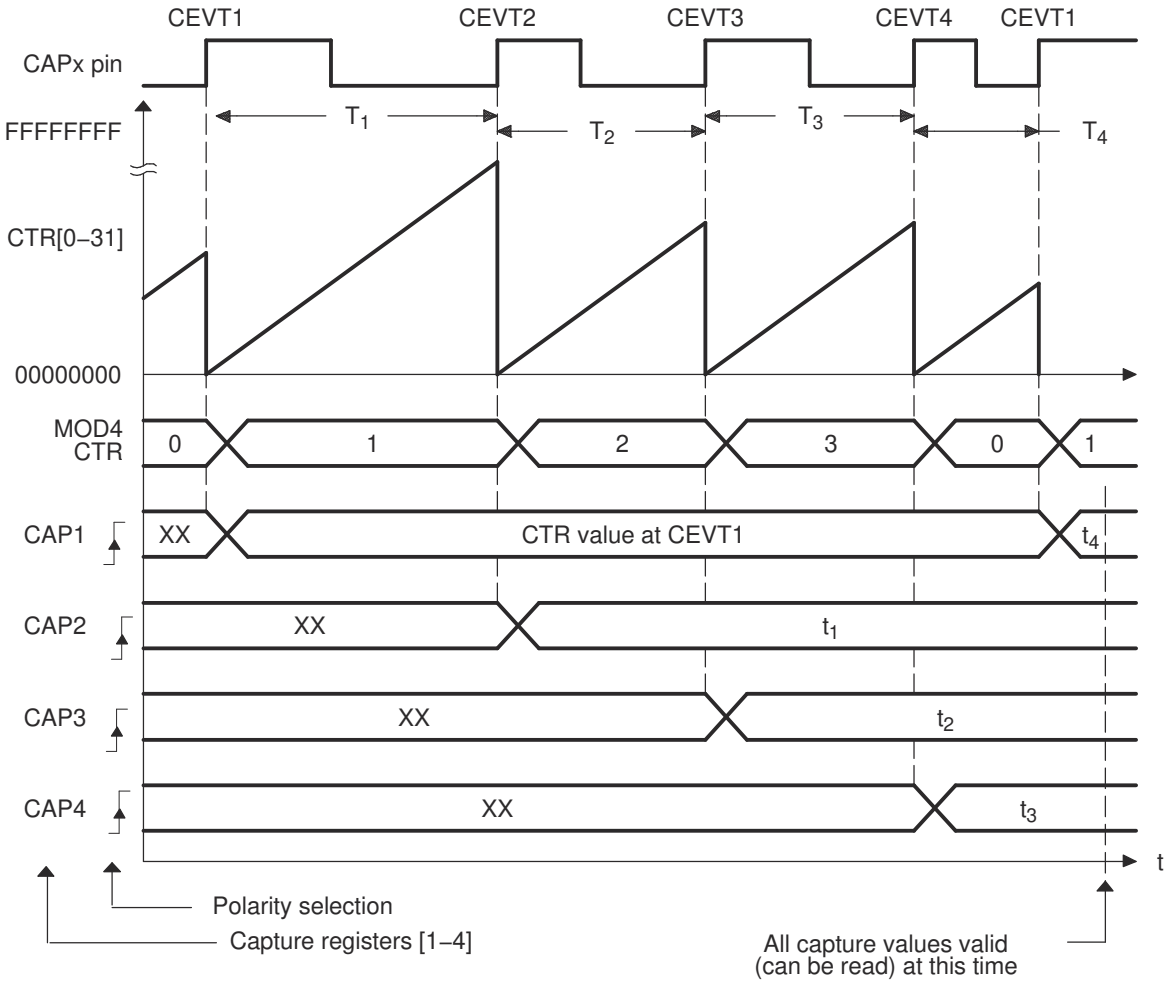


Figure 28-16. Capture Sequence for Delta Mode Time-stamp and Rising Edge Detect

28.6.4 Example 4 - Time Difference (Delta) Operation Rising- and Falling-Edge Trigger

In Figure 28-17, the eCAP operating mode is almost the same as in previous section except capture events are qualified as either rising or falling edge, this now gives both period and duty cycle information, that is: Period1 = T_1+T_2, Period2 = T_3+T_4, and so on. Duty Cycle1 (on-time %) = T_1 / Period1 x 100%, Duty Cycle1 (off-time %) = T_2 / Period1 x 100%, and so on.

During initialization, write to the active registers for both period and compare. This action automatically copies the init values into the shadow values. For subsequent compare updates during run-time, the shadow registers must be used.

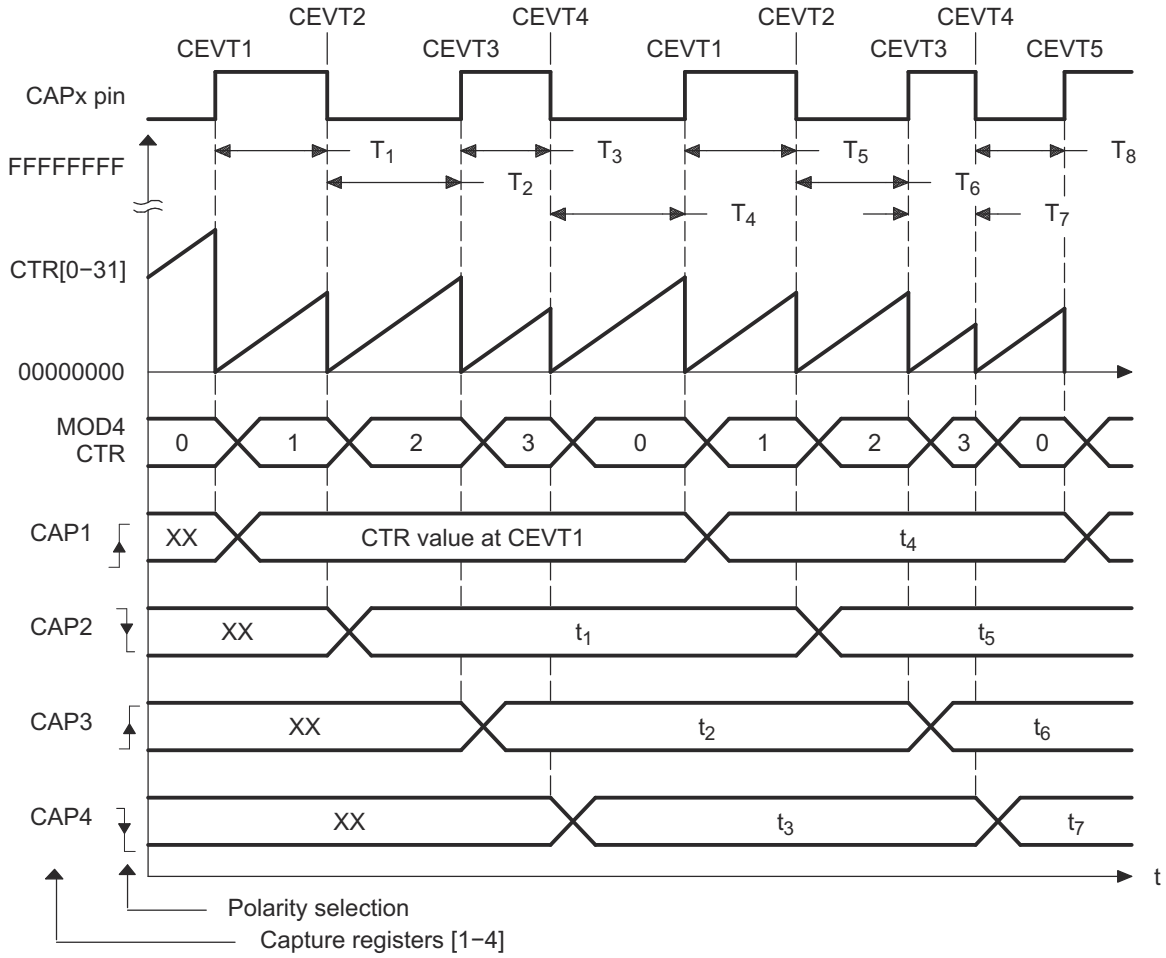


Figure 28-17. Capture Sequence for Delta Mode Time-stamp with Rising- and Falling-Edge Detect

28.7 Application of the APWM Mode

In this example, the eCAP module is configured to operate as a PWM generator. Here, a very simple single-channel PWM waveform is generated from the APWMx output pin. The PWM polarity is active high, which means that the compare value (CAP2 reg is now a compare register) represents the on-time (high level) of the period. Alternatively, if the APWMPOL bit is configured for active low, then the compare value represents the off-time.

28.7.1 Example 1 - Simple PWM Generation (Independent Channels)

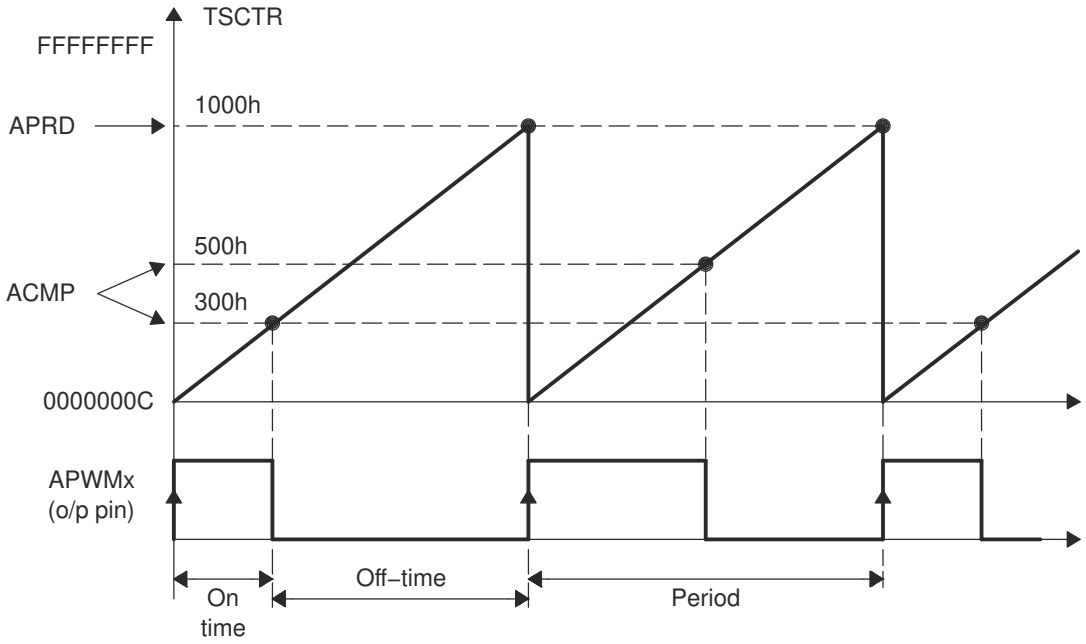


Figure 28-18. PWM Waveform Details of APWM Mode Operation

28.8 Software

28.8.1 ECAP Registers to Driverlib Functions

Table 28-2. ECAP Registers to Driverlib Functions

File	Driverlib Function
TSCTR	
ecap.h	ECAP_getTimeBaseCounter
CTRPHS	
ecap.h	ECAP_setPhaseShiftCount
CAP1	
ecap.h	ECAP_setAPWMPeriod
ecap.h	ECAP_getEventTimeStamp
CAP2	
ecap.h	ECAP_setAPWMCompare
ecap.h	ECAP_getEventTimeStamp
CAP3	
ecap.h	ECAP_setAPWMShadowPeriod
ecap.h	ECAP_getEventTimeStamp
CAP4	
ecap.h	ECAP_setAPWMShadowCompare
ecap.h	ECAP_getEventTimeStamp
ECCTL0	
ecap.h	ECAP_selectECAPInput
ecap.h	ECAP_selectQualPeriod
ecap.h	ECAP_setSOCTriggerSource
ECCTL1	
ecap.c	ECAP_setEmulationMode
ecap.h	ECAP_setEventPrescaler
ecap.h	ECAP_setEventPolarity
ecap.h	ECAP_enableCounterResetOnEvent
ecap.h	ECAP_disableCounterResetOnEvent
ecap.h	ECAP_enableTimeStampCapture
ecap.h	ECAP_disableTimeStampCapture
ECCTL2	
ecap.h	ECAP_setCaptureMode
ecap.h	ECAP_reArm
ecap.h	ECAP_enableCaptureMode
ecap.h	ECAP_enableAPWMMode
ecap.h	ECAP_enableLoadCounter
ecap.h	ECAP_disableLoadCounter
ecap.h	ECAP_loadCounter
ecap.h	ECAP_setSyncOutMode
ecap.h	ECAP_stopCounter
ecap.h	ECAP_startCounter
ecap.h	ECAP_setAPWMPolarity
ecap.h	ECAP_resetCounters
ecap.h	ECAP_setDMASource

Table 28-2. ECAP Registers to Driverlib Functions (continued)

File	Driverlib Function
ecap.h	ECAP_getModuloCounterStatus
ECEINT	
ecap.h	ECAP_enableInterrupt
ecap.h	ECAP_disableInterrupt
ECFLG	
ecap.h	ECAP_getInterruptSource
ecap.h	ECAP_getGlobalInterruptStatus
ECCLR	
ecap.h	ECAP_clearInterrupt
ecap.h	ECAP_clearGlobalInterrupt
ECFRC	
ecap.h	ECAP_forceInterrupt
SYNCINSEL	
ecap.h	ECAP_setSyncInPulseSource
HRCTL	
hrcap.h	HRCAP_enableHighResolution
hrcap.h	HRCAP_disableHighResolution
hrcap.h	HRCAP_enableHighResolutionClock
hrcap.h	HRCAP_disbleHighResolutionClock
hrcap.h	HRCAP_startCalibration
hrcap.h	HRCAP_setCalibrationMode
hrcap.h	HRCAP_isCalibrationBusy
HRINTEN	
hrcap.h	HRCAP_enableCalibrationInterrupt
hrcap.h	HRCAP_disableCalibrationInterrupt
HRFLG	
hrcap.h	HRCAP_getCalibrationFlags
HRCLR	
hrcap.h	HRCAP_clearCalibrationFlags
HRFRC	
hrcap.h	HRCAP_forceCalibrationFlags
HRCALPRD	
hrcap.h	HRCAP_setCalibrationPeriod
hrcap.h	HRCAP_configCalibrationPeriod
HRSYSCLKCTR	
-	
HRSYSCLKCAP	
hrcap.h	HRCAP_getCalibrationClockPeriod
HRCLKCTR	
-	
HRCLKCAP	
hrcap.h	HRCAP_getCalibrationClockPeriod
MUNIT_COMMON_CTL	
ecap.h	ECAP_selectTripSignal
ecap.h	ECAP_selectGlobalLoadStrobe

Table 28-2. ECAP Registers to Driverlib Functions (continued)

File	Driverlib Function
MUNIT_1_CTL	
ecap.h	ECAP_enableSignalMonitoringUnit
ecap.h	ECAP_disableSignalMonitoringUnit
ecap.h	ECAP_enableDebugRange
ecap.h	ECAP_disableDebugRange
ecap.h	ECAP_setupEarlyMaxErrorCheck
ecap.h	ECAP_selectMonitoringType
MUNIT_1_SHADOW_CTL	
ecap.h	ECAP_enableShadowMinMaxRegisters
ecap.h	ECAP_disableShadowMinMaxRegisters
ecap.h	ECAP_enableSoftwareSync
ecap.h	ECAP_selectShadowLoadMode
MUNIT_1_MIN	
ecap.h	ECAP_configureMinValue
ecap.h	ECAP_configureShadowMinValue
MUNIT_1_MAX	
ecap.h	ECAP_configureMaxValue
ecap.h	ECAP_configureShadowMaxValue
MUNIT_1_MIN_SHADOW	
ecap.h	ECAP_configureShadowMinValue
MUNIT_1_MAX_SHADOW	
ecap.h	ECAP_configureShadowMaxValue
MUNIT_1_DEBUG_RANGE_MIN	
ecap.h	ECAP_observedMinValue
MUNIT_1_DEBUG_RANGE_MAX	
ecap.h	ECAP_observedMaxValue
MUNIT_2_CTL	
-	
MUNIT_2_SHADOW_CTL	
-	
MUNIT_2_MIN	
-	
MUNIT_2_MAX	
-	
MUNIT_2_MIN_SHADOW	
-	
MUNIT_2_MAX_SHADOW	
-	
MUNIT_2_DEBUG_RANGE_MIN	
-	
MUNIT_2_DEBUG_RANGE_MAX	
-	

28.8.2 ECAP Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/ecap

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

28.8.2.1 eCAP APWM Example - SINGLE_CORE

FILE: ecap_ex1_apwm.c

This program sets up the eCAP module in APWM mode. The PWM waveform will come out on GPIO5. The frequency of PWM is configured to vary between 10Hz and 20Hz using the shadow registers to load the next period/compare values.

28.8.2.2 eCAP Capture PWM Example - SINGLE_CORE

FILE: ecap_ex2_capture_pwm.c

This example configures ePWM3A for:

- Up count mode
- Period starts at 500 and goes up to 8000
- Toggle output on PRD

eCAP1 is configured to capture the time between rising and falling edge of the ePWM3A output.

External Connections

- eCAP1 is on GPIO16
- ePWM1A is on GPIO4
- Connect GPIO4 to GPIO16.

Watch Variables

- *ecap1PassCount* - Successful captures.
- *ecap1IntCount* - Interrupt counts.

28.8.2.3 eCAP APWM Phase-shift Example - SINGLE_CORE

FILE: ecap_ex3_apwm_phase_shift.c

This program sets up the eCAP1 and eCAP2 modules in APWM mode to generate the two phase-shifted PWM outputs of same duty and frequency value The frequency, duty and phase values can be programmed of choice by updating the defined macros. By default 10 Khz frequency, 50% duty and 30% phase shift values are used. eCAP2 output leads the eCAP1 output by 30% GPIO0 and GPIO1 are used as eCAP1/2 outputs and can be probed using analyzer/CRO to observe the waveforms.

28.9 ECAP Registers

This section describes the ECAP Registers.

28.9.1 ECAP Base Address Table

Table 28-3. ECAP Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
ECAP_REGS	ECAP1_BASE	0x7010_0000	YES	YES	YES	YES	YES	YES	-	YES
ECAP_SIGNAL_MONITORING	ECAP1SIGNALMONITORING_BASE	0x7010_0080	YES	YES	YES	YES	YES	YES	-	YES
ECAP_REGS	ECAP2_BASE	0x7010_1000	YES	YES	YES	YES	YES	YES	-	YES
ECAP_SIGNAL_MONITORING	ECAP2SIGNALMONITORING_BASE	0x7010_1080	YES	YES	YES	YES	YES	YES	-	YES
ECAP_REGS	ECAP3_BASE	0x7010_2000	YES	YES	YES	YES	YES	YES	-	YES
ECAP_SIGNAL_MONITORING	ECAP3SIGNALMONITORING_BASE	0x7010_2080	YES	YES	YES	YES	YES	YES	-	YES
ECAP_REGS	ECAP4_BASE	0x7010_3000	YES	YES	YES	YES	YES	YES	-	YES
ECAP_SIGNAL_MONITORING	ECAP4SIGNALMONITORING_BASE	0x7010_3080	YES	YES	YES	YES	YES	YES	-	YES
ECAP_REGS	ECAP5_BASE	0x7010_4000	YES	YES	YES	YES	YES	YES	-	YES
HRCAP_REGS	HRCAP5_BASE	0x7010_4040	YES	YES	YES	YES	YES	YES	-	YES
ECAP_SIGNAL_MONITORING	ECAP5SIGNALMONITORING_BASE	0x7010_4080	YES	YES	YES	YES	YES	YES	-	YES
ECAP_REGS	ECAP6_BASE	0x7010_5000	YES	YES	YES	YES	YES	YES	-	YES
HRCAP_REGS	HRCAP6_BASE	0x7010_5040	YES	YES	YES	YES	YES	YES	-	YES
ECAP_SIGNAL_MONITORING	ECAP6SIGNALMONITORING_BASE	0x7010_5080	YES	YES	YES	YES	YES	YES	-	YES

28.9.2 ECAP_REGS Registers

Table 28-4 lists the memory-mapped registers for the ECAP_REGS registers. All register offset addresses not listed in Table 28-4 should be considered as reserved locations and the register contents should not be modified.

Table 28-4. ECAP_REGS Registers

Offset	Acronym	Register Name	Protection
0h	TSCTR	Time-Stamp Counter	
4h	CTRPHS	Counter Phase Offset Value Register	
8h	CAP1	Capture 1 Register	
Ch	CAP2	Capture 2 Register	
10h	CAP3	Capture 3 Register	
14h	CAP4	Capture 4 Register	
24h	ECCTL0	Capture Control Register 0	
28h	ECCTL1	Capture Control Register 1	
2Ah	ECCTL2	Capture Control Register 2	
2Ch	ECEINT	Capture Interrupt Enable Register	
2Eh	ECFLG	Capture Interrupt Flag Register	
30h	ECCLR	Capture Interrupt Clear Register	
32h	ECFRC	Capture Interrupt Force Register	
3Ch	ECAPSYNCINSEL	SYNC source select register	

Complex bit access types are encoded to fit into small table cells. Table 28-5 shows the codes that are used for access types in this section.

Table 28-5. ECAP_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

28.9.2.1 TSCTR Register (Offset = 0h) [Reset = 00000000h]

TSCTR is shown in Figure 28-19 and described in Table 28-6.

Return to the Summary Table.

Time-Stamp Counter

Figure 28-19. TSCTR Register

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

Table 28-6. TSCTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TSCTR	R/W	0h	Active 32-bit counter register that is used as the capture time-base HR mode : 1) This register reads HRCOUNTER value and is not writable 2) can be reset using CTRFILTRESET 3) Its not synchronized to SYSCLK domain so reads may not be accurate Reset type: SYSRSn

28.9.2.2 CTRPHS Register (Offset = 4h) [Reset = 00000000h]

CTRPHS is shown in Figure 28-20 and described in Table 28-7.

Return to the Summary Table.

Counter Phase Offset Value Register

Figure 28-20. CTRPHS Register

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

Table 28-7. CTRPHS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CTRPHS	R/W	0h	Counter phase value register that can be programmed for phase lag/lead. This register CTRPHS is loaded into TSCTR upon either a SYNCI event or S/W force via a control bit. Used to achieve phase control synchronization with respect to other eCAP and EPWM time-bases. This register is not applicable in HR mode. Reset type: SYSRSn

28.9.2.3 CAP1 Register (Offset = 8h) [Reset = 00000000h]

CAP1 is shown in Figure 28-21 and described in Table 28-8.

Return to the Summary Table.

Capture 1 Register

Figure 28-21. CAP1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP1																															
R/W-0h																															

Table 28-8. CAP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP1	R/W	0h	This register can be loaded (written) by: - Time-Stamp counter value (TSCTR) during a capture event - Software - may be useful for test purposes or initialization - ARPD shadow register (CAP3) when used in APWM mode Reset type: SYSRSn

28.9.2.4 CAP2 Register (Offset = Ch) [Reset = 00000000h]

CAP2 is shown in Figure 28-22 and described in Table 28-9.

Return to the Summary Table.

Capture 2 Register

Figure 28-22. CAP2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP2																															
R/W-0h																															

Table 28-9. CAP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP2	R/W	0h	This register can be loaded (written) by: - Time-Stamp (counter value) during a capture event - Software - may be useful for test purposes - ACMP shadow register (CAP4) when used in APWM mode Reset type: SYSRSn

28.9.2.5 CAP3 Register (Offset = 10h) [Reset = 00000000h]

CAP3 is shown in Figure 28-23 and described in Table 28-10.

Return to the Summary Table.

Capture 3 Register

Figure 28-23. CAP3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP3																															
R/W-0h																															

Table 28-10. CAP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP3	R/W	0h	In CMP mode, this is a time-stamp capture register. In APWM mode, this is the period shadow (APRD) register. You can update the PWM period value through this register. CAP3 (APRD) shadows CAP1 in this mode. Reset type: SYSRSn

28.9.2.6 CAP4 Register (Offset = 14h) [Reset = 00000000h]

CAP4 is shown in Figure 28-24 and described in Table 28-11.

Return to the Summary Table.

Capture 4 Register

Figure 28-24. CAP4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP4																															
R/W-0h																															

Table 28-11. CAP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP4	R/W	0h	In CMP mode, this is a time-stamp capture register. In APWM mode, this is the compare shadow (ACMP) register. You can update the PWM compare value via this register. CAP4 (ACMP) shadows CAP2 in this mode. Reset type: SYSRSn

28.9.2.7 ECCTL0 Register (Offset = 24h) [Reset = 000000FFh]

ECCTL0 is shown in Figure 28-25 and described in Table 28-12.

Return to the Summary Table.

Capture Control Register 0

Figure 28-25. ECCTL0 Register

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

Table 28-12. ECCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	Reserved
17-16	SOCEVTSEL	R/W	0h	ADC SOC event select Capture Mode: 00b (R/W) = SOC trigger source is CEVT1 01b (R/W) = SOC trigger source is CEVT2 10b (R/W) = SOC trigger source is CEVT3 11b (R/W) = SOC trigger source is CEVT4 APWM Mode: 00b (R/W) = SOC trigger interrupt source is period match 01b (R/W) = SOC trigger interrupt source is compare match 10b (R/W) = SOC trigger interrupt source is period match or compare match 11b (R/W) = Disabled Reset type: CPU1.SYSRSn
15-12	QUALPRD	R/W	0h	Qual period to filter out noise on input signals being monitored, Not applicable for HR mode. 0x0 : Bypass 0x1 : pulses of with 1 cycle or less will be filtered out 0x2 : pulses of with 2 cycles or less will be filtered out .... 0xF : pulses of with 15 cycles or less will be filtered out Reset type: CPU1.SYSRSn
11-8	RESERVED	R-0	0h	Reserved
7-0	INPUTSEL	R/W	FFh	Capture input source select bits 0x0 capture input is ECAPxINPUT[0] 0x1 capture input is ECAPxINPUT[1] 0x2 capture input is ECAPxINPUT[2] ... 0xFF capture input is ECAPxINPUT[256] Reset type: CPU1.SYSRSn

28.9.2.8 ECCTL1 Register (Offset = 28h) [Reset = 0000h]

ECCTL1 is shown in Figure 28-26 and described in Table 28-13.

Return to the Summary Table.

Capture Control Register 1

Figure 28-26. ECCTL1 Register

15	14	13	12	11	10	9	8
FREE_SOFT		PRESCALE					CAPLDEN
R/W-0h		R/W-0h					R/W-0h
7	6	5	4	3	2	1	0
CTRRST4	CAP4POL	CTRRST3	CAP3POL	CTRRST2	CAP2POL	CTRRST1	CAP1POL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 28-13. ECCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	FREE_SOFT	R/W	0h	Emulation Control Reset type: SYSRSn 0h (R/W) = TSCTR counter stops immediately on emulation suspend 1h (R/W) = TSCTR counter runs until = 0 2h (R/W) = TSCTR counter is unaffected by emulation suspend (Run Free) 3h (R/W) = TSCTR counter is unaffected by emulation suspend (Run Free)
13-9	PRESCALE	R/W	0h	Event Filter prescale select Reset type: SYSRSn 0h (R/W) = Divide by 1 (i.e,. no prescale, by-pass the prescaler) 1h (R/W) = Divide by 2 2h (R/W) = Divide by 4 3h (R/W) = Divide by 6 4h (R/W) = Divide by 8 5h (R/W) = Divide by 10 1Eh (R/W) = Divide by 60 1Fh (R/W) = Divide by 62
8	CAPLDEN	R/W	0h	Enable Loading of CAP1-4 registers on a capture event. Note that this bit does not disable CEVTn events from being generated. Reset type: SYSRSn 0h (R/W) = Disable CAP1-4 register loads at capture event time. 1h (R/W) = Enable CAP1-4 register loads at capture event time.
7	CTRRST4	R/W	0h	Counter Reset on Capture Event 4 Reset type: SYSRSn 0h (R/W) = Do not reset counter on Capture Event 4 (absolute time stamp operation) 1h (R/W) = Reset counter after Capture Event 4 time-stamp has been captured (used in difference mode operation)
6	CAP4POL	R/W	0h	Capture Event 4 Polarity select Reset type: SYSRSn 0h (R/W) = Capture Event 4 triggered on a rising edge (RE) 1h (R/W) = Capture Event 4 triggered on a falling edge (FE)
5	CTRRST3	R/W	0h	Counter Reset on Capture Event 3 Reset type: SYSRSn 0h (R/W) = Do not reset counter on Capture Event 3 (absolute time stamp) 1h (R/W) = Reset counter after Event 3 time-stamp has been captured (used in difference mode operation)

Table 28-13. ECCTL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CAP3POL	R/W	0h	Capture Event 3 Polarity select Reset type: SYSRSn 0h (R/W) = Capture Event 3 triggered on a rising edge (RE) 1h (R/W) = Capture Event 3 triggered on a falling edge (FE)
3	CTRRST2	R/W	0h	Counter Reset on Capture Event 2 Reset type: SYSRSn 0h (R/W) = Do not reset counter on Capture Event 2 (absolute time stamp) 1h (R/W) = Reset counter after Event 2 time-stamp has been captured (used in difference mode operation)
2	CAP2POL	R/W	0h	Capture Event 2 Polarity select Reset type: SYSRSn 0h (R/W) = Capture Event 2 triggered on a rising edge (RE) 1h (R/W) = Capture Event 2 triggered on a falling edge (FE)
1	CTRRST1	R/W	0h	Counter Reset on Capture Event 1 Reset type: SYSRSn 0h (R/W) = Do not reset counter on Capture Event 1 (absolute time stamp) 1h (R/W) = Reset counter after Event 1 time-stamp has been captured (used in difference mode operation)
0	CAP1POL	R/W	0h	Capture Event 1 Polarity select Reset type: SYSRSn 0h (R/W) = Capture Event 1 triggered on a rising edge (RE) 1h (R/W) = Capture Event 1 triggered on a falling edge (FE)

28.9.2.9 ECCTL2 Register (Offset = 2Ah) [Reset = 0006h]

ECCTL2 is shown in Figure 28-27 and described in Table 28-14.

Return to the Summary Table.

Capture Control Register 2

Figure 28-27. ECCTL2 Register

15	14	13	12	11	10	9	8
MODCNTRSTS		DMAEVTSEL		CTRFILTRESET	APWMPOL	CAP_APWM	SWSYNC
R/W-0h		R/W-0h		R-0/W1C-0h	R/W-0h	R/W-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
SYNCO_SEL		SYNCI_EN	TSCTRSTOP	REARM	STOP_WRAP		CONT_ONESHT
R/W-0h		R/W-0h	R/W-0h	R-0/W1S-0h	R/W-3h		R/W-0h

Table 28-14. ECCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	MODCNTRSTS	R/W	0h	This bit field reads current status on modulo counter 00b (R) = CAP1 register gets loaded on next capture event. 01b (R) = CAP2 register gets loaded on next capture event. 10b (R) = CAP3 register gets loaded on next capture event. 11b (R) = CAP4 register gets loaded on next capture event. Reset type: CPU1.SYSRSn
13-12	DMAEVTSEL	R/W	0h	DMA event select Capture Mode: 00b (R/W) = DMA interrupt source is CEVT1 01b (R/W) = DMA interrupt source is CEVT2 10b (R/W) = DMA interrupt source is CEVT3 11b (R/W) = DMA interrupt source is CEVT4 APWM Mode: 00b (R/W) = DMA interrupt source is period match 01b (R/W) = DMA interrupt source is compare match 10b (R/W) = DMA interrupt source is period match or compare match 11b (R/W) = Disabled Reset type: CPU1.SYSRSn
11	CTRFILTRESET	R-0/W1C	0h	Reset Bit 0h (R) = No effect 1h (W) = Resets event filter, counter, modulo counter and CEVT[1,2,3,4] and CNTOVF , HRERROR flags Note: This provides an ability start capture module from known state in case spurious inputs are captured while ECAP is configured. Reset type: CPU1.SYSRSn
10	APWMPOL	R/W	0h	APWM output polarity select. This is applicable only in APWM operating mode. Reset type: SYSRSn 0h (R/W) = Output is active high (Compare value defines high time) 1h (R/W) = Output is active low (Compare value defines low time)

Table 28-14. ECCTL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	CAP_APWM	R/W	0h	CAP/APWM operating mode select Reset type: SYSRSn 0h (R/W) = ECAP module operates in capture mode. This mode forces the following configuration: - Inhibits TSCTR resets via CTR = PRD event - Inhibits shadow loads on CAP1 and 2 registers - Permits user to enable CAP1-4 register load - CAPx/APWMx pin operates as a capture input 1h (R/W) = ECAP module operates in APWM mode. This mode forces the following configuration: - Resets TSCTR on CTR = PRD event (period boundary - Permits shadow loading on CAP1 and 2 registers - Disables loading of time-stamps into CAP1-4 registers - CAPx/APWMx pin operates as a APWM output
8	SWSYNC	R-0/W1S	0h	Software-forced Counter (TSCTR) Synchronizer. This provides the user a method to generate a synchronization pulse through software. In APWM mode, the synchronization pulse can also be sourced from the CTR = PRD event. Reset type: SYSRSn 0h (R/W) = Writing a zero has no effect. Reading always returns a zero 1h (R/W) = Writing a one forces a TSCTR shadow load of current ECAP module and any ECAP modules down-stream providing the SYNCO_SEL bits are 0,0. After writing a 1, this bit returns to a zero.
7-6	SYNCO_SEL	R/W	0h	Sync-Out Select Reset type: SYSRSn 0h (R/W) = sync out signal is SWSYNC 1h (R/W) = Select CTR = PRD event to be the sync-out signal. Note: Selection CTR = PRD is meaningful only in APWM mode 2h (R/W) = Disable sync out signal 3h (R/W) = Disable sync out signal
5	SYNCI_EN	R/W	0h	Counter (TSCTR) Sync-In select mode Reset type: SYSRSn 0h (R/W) = Disable sync-in option 1h (R/W) = Enable counter (TSCTR) to be loaded from CTRPHS register upon either a SYNCI signal or a S/W force event.
4	TSCTRSTOP	R/W	0h	Time Stamp (TSCTR) Counter Stop (freeze) Control Reset type: SYSRSn 0h (R/W) = TSCTR stopped 1h (R/W) = TSCTR free-running
3	REARM	R-0/W1S	0h	Re-Arming Control. Note: The re-arm function is valid in one shot or continuous mode Reset type: SYSRSn 0h (R/W) = Has no effect (reading always returns a 0) 1h (R/W) = Arms the one-shot sequence as follows: 1) Resets the Mod4 counter to zero 2) Unfreezes the Mod4 counter 3) Enables capture register loads

Table 28-14. ECCTL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-1	STOP_WRAP	R/W	3h	Stop value for one-shot mode. This is the number (between 1-4) of captures allowed to occur before the CAP(1-4) registers are frozen, that is, capture sequence is stopped. Wrap value for continuous mode. This is the number (between 1-4) of the capture register in which the circular buffer wraps around and starts again. Notes: STOP_WRAP is compared to Mod4 counter and, when equal, 2 actions occur: - Mod4 counter is stopped (frozen) - Capture register loads are inhibited In one-shot mode, further interrupt events are blocked until re-armed. Reset type: SYSRSn 0h (R/W) = Stop after Capture Event 1 in one-shot mode Wrap after Capture Event 1 in continuous mode. 1h (R/W) = Stop after Capture Event 2 in one-shot mode Wrap after Capture Event 2 in continuous mode. 2h (R/W) = Stop after Capture Event 3 in one-shot mode Wrap after Capture Event 3 in continuous mode. 3h (R/W) = Stop after Capture Event 4 in one-shot mode Wrap after Capture Event 4 in continuous mode.
0	CONT_ONESHT	R/W	0h	Continuous or one-shot mode control (applicable only in capture mode) Reset type: SYSRSn 0h (R/W) = Operate in continuous mode 1h (R/W) = Operate in one-Shot mode

28.9.2.10 ECEINT Register (Offset = 2Ch) [Reset = 0000h]

ECEINT is shown in Figure 28-28 and described in Table 28-15.

Return to the Summary Table.

The interrupt enable bits (CEVT1, ...) block any of the selected events from generating an interrupt. Events will still be latched into the flag bit (ECFLG register) and can be forced/cleared via the ECFRC/ECCLR registers.
The proper procedure for configuring peripheral modes and interrupts is as follows:
- Disable global interrupts
- Stop eCAP counter
- Disable eCAP interrupts
- Configure peripheral registers
- Clear spurious eCAP interrupt flags
- Enable eCAP interrupts
- Start eCAP counter
- Enable global interrupts

Figure 28-28. ECEINT Register

15	14	13	12	11	10	9	8
RESERVED			MUNIT_2_ERROR_EVT2	MUNIT_2_ERROR_EVT1	MUNIT_1_ERROR_EVT2	MUNIT_1_ERROR_EVT1	RESERVED
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CTR_EQ_CMP	CTR_EQ_PRD	CTROVF	CEVT4	CEVT3	CEVT2	CEVT1	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 28-15. ECEINT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	MUNIT_2_ERROR_EVT2	R/W	0h	Monitoring unit 2 error event 2 interrupt enable 0 : Disable Monitoring unit 2 error event 2 interrupt 1 : Enable Monitoring unit 2 error event 2 interrupt Reset type: CPU1.SYSRSn
11	MUNIT_2_ERROR_EVT1	R/W	0h	Monitoring unit 2 error event 2 interrupt enable 0 : Disable Monitoring unit 2 error event 1 interrupt 1 : Enable Monitoring unit 2 error event 1 interrupt Reset type: CPU1.SYSRSn
10	MUNIT_1_ERROR_EVT2	R/W	0h	Monitoring unit 1 error event 1 interrupt enable 0 : Disable Monitoring unit 1 error event 2 interrupt 1 : Enable Monitoring unit 1 error event 2 interrupt Reset type: CPU1.SYSRSn
9	MUNIT_1_ERROR_EVT1	R/W	0h	Monitoring unit 1 error event 1 interrupt enable 0 : Disable Monitoring unit 1 error event 1 interrupt 1 : Enable Monitoring unit 1 error event 1 interrupt Reset type: CPU1.SYSRSn
8	RESERVED	R/W	0h	Reserved
7	CTR_EQ_CMP	R/W	0h	Counter Equal Compare Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Compare Equal as an Interrupt source 1h (R/W) = Enable Compare Equal as an Interrupt source
6	CTR_EQ_PRD	R/W	0h	Counter Equal Period Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Period Equal as an Interrupt source 1h (R/W) = Enable Period Equal as an Interrupt source

Table 28-15. ECEINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	CTROVF	R/W	0h	Counter Overflow Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disabled counter Overflow as an Interrupt source 1h (R/W) = Enable counter Overflow as an Interrupt source
4	CEVT4	R/W	0h	Capture Event 4 Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Capture Event 4 as an Interrupt source 1h (R/W) = Capture Event 4 Interrupt Enable
3	CEVT3	R/W	0h	Capture Event 3 Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Capture Event 3 as an Interrupt source 1h (R/W) = Enable Capture Event 3 as an Interrupt source
2	CEVT2	R/W	0h	Capture Event 2 Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Capture Event 2 as an Interrupt source 1h (R/W) = Enable Capture Event 2 as an Interrupt source
1	CEVT1	R/W	0h	Capture Event 1 Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Capture Event 1 as an Interrupt source 1h (R/W) = Enable Capture Event 1 as an Interrupt source
0	RESERVED	R	0h	Reserved

28.9.2.11 ECFLG Register (Offset = 2Eh) [Reset = 0000h]

ECFLG is shown in Figure 28-29 and described in Table 28-16.

Return to the Summary Table.

Capture Interrupt Flag Register

Figure 28-29. ECFLG Register

15	14	13	12	11	10	9	8
RESERVED			MUNIT_2_ERROR_EVT2	MUNIT_2_ERROR_EVT1	MUNIT_1_ERROR_EVT2	MUNIT_1_ERROR_EVT1	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
CTR_CMP	CTR_PRD	CTROVF	CEVT4	CEVT3	CEVT2	CEVT1	INT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 28-16. ECFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	MUNIT_2_ERROR_EVT2	R	0h	Error event 2 Interrupt Flag from monitoring unit 2 Reset type: SYSRSn
11	MUNIT_2_ERROR_EVT1	R	0h	Error event 2 Interrupt Flag from monitoring unit 2 Reset type: SYSRSn
10	MUNIT_1_ERROR_EVT2	R	0h	Error event 2 Interrupt Flag from monitoring unit 1 Reset type: SYSRSn
9	MUNIT_1_ERROR_EVT1	R	0h	Error event 2 Interrupt Flag from monitoring unit 1 Reset type: SYSRSn
8	RESERVED	R	0h	Reserved
7	CTR_CMP	R	0h	Compare Equal Compare Status Flag. This flag is active only in APWM mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the counter (TSCTR) reached the compare register value (ACMP)
6	CTR_PRD	R	0h	Counter Equal Period Status Flag. This flag is only active in APWM mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the counter (TSCTR) reached the period register value (APRD) and was reset.
5	CTROVF	R	0h	Counter Overflow Status Flag. This flag is active in CAP and APWM mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the counter (TSCTR) has made the transition from FFFFFFFF to 00000000
4	CEVT4	R	0h	Capture Event 4 Status Flag This flag is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the fourth event occurred at ECAPx pin
3	CEVT3	R	0h	Capture Event 3 Status Flag. This flag is active only in CAP mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the third event occurred at ECAPx pin.

Table 28-16. ECFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	CEVT2	R	0h	Capture Event 2 Status Flag. This flag is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the second event occurred at ECAPx pin.
1	CEVT1	R	0h	Capture Event 1 Status Flag. This flag is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the first event occurred at ECAPx pin.
0	INT	R	0h	Global Interrupt Status Flag Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates that an interrupt was generated.

28.9.2.12 ECCLR Register (Offset = 30h) [Reset = 0000h]

ECCLR is shown in Figure 28-30 and described in Table 28-17.

Return to the Summary Table.

Capture Interrupt Clear Register

Figure 28-30. ECCLR Register

15	14	13	12	11	10	9	8
RESERVED			MUNIT_2_ERROR_EVT2	MUNIT_2_ERROR_EVT1	MUNIT_1_ERROR_EVT2	MUNIT_1_ERROR_EVT1	RESERVED
R-0h			R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
CTR_CMP	CTR_PRD	CTROVF	CEVT4	CEVT3	CEVT2	CEVT1	INT
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 28-17. ECCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	MUNIT_2_ERROR_EVT2	R-0/W1C	0h	Writing '1' clears MUNIT_2_ERROR_EVT2 interrupt flag Reset type: SYSRSn
11	MUNIT_2_ERROR_EVT1	R-0/W1C	0h	Writing '1' clears MUNIT_2_ERROR_EVT1 interrupt flag Reset type: SYSRSn
10	MUNIT_1_ERROR_EVT2	R-0/W1C	0h	Writing '1' clears MUNIT_1_ERROR_EVT2 interrupt flag Reset type: SYSRSn
9	MUNIT_1_ERROR_EVT1	R-0/W1C	0h	Writing '1' clears MUNIT_1_ERROR_EVT1 interrupt flag Reset type: SYSRSn
8	RESERVED	R-0/W1C	0h	Reserved
7	CTR_CMP	R-0/W1C	0h	Counter Equal Compare Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CTR=CMP flag.
6	CTR_PRD	R-0/W1C	0h	Counter Equal Period Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CTR=PRD flag.
5	CTROVF	R-0/W1C	0h	Counter Overflow Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CTROVF flag.
4	CEVT4	R-0/W1C	0h	Capture Event 4 Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CEVT4 flag.
3	CEVT3	R-0/W1C	0h	Capture Event 3 Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CEVT3 flag.
2	CEVT2	R-0/W1C	0h	Capture Event 2 Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CEVT2 flag.

Table 28-17. ECCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	CEVT1	R-0/W1C	0h	Capture Event 1 Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CEVT1 flag.
0	INT	R-0/W1C	0h	ECAP Global Interrupt Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the INT flag and enable further interrupts to be generated if any of the event flags are set to 1

28.9.2.13 ECFRC Register (Offset = 32h) [Reset = 0000h]

ECFRC is shown in Figure 28-31 and described in Table 28-18.

Return to the Summary Table.

Capture Interrupt Force Register

Figure 28-31. ECFRC Register

15	14	13	12	11	10	9	8
RESERVED			MUNIT_2_ERROR_EVT2	MUNIT_2_ERROR_EVT1	MUNIT_1_ERROR_EVT2	MUNIT_1_ERROR_EVT1	RESERVED
R-0h			R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
CTR_CMP	CTR_PRD	CTROVF	CEVT4	CEVT3	CEVT2	CEVT1	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h

Table 28-18. ECFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	MUNIT_2_ERROR_EVT2	R-0/W1S	0h	Writing '1' sets MUNIT_2_ERROR_EVT2 interrupt flag Reset type: SYSRSn
11	MUNIT_2_ERROR_EVT1	R-0/W1S	0h	Writing '1' sets MUNIT_2_ERROR_EVT1 interrupt flag Reset type: SYSRSn
10	MUNIT_1_ERROR_EVT2	R-0/W1S	0h	Writing '1' sets MUNIT_1_ERROR_EVT2 interrupt flag Reset type: SYSRSn
9	MUNIT_1_ERROR_EVT1	R-0/W1S	0h	Writing '1' sets MUNIT_1_ERROR_EVT1 interrupt flag Reset type: SYSRSn
8	RESERVED	R-0/W1S	0h	Reserved
7	CTR_CMP	R-0/W1S	0h	Force Counter Equal Compare Interrupt. This event is only active in APWM mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CTR_CMP flag.
6	CTR_PRD	R-0/W1S	0h	Force Counter Equal Period Interrupt. This event is only active in APWM mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CTR_PRD flag.
5	CTROVF	R-0/W1S	0h	Force Counter Overflow Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 to this bit sets the CTROVF flag.
4	CEVT4	R-0/W1S	0h	Force Capture Event 4. This event is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CEVT4 flag.
3	CEVT3	R-0/W1S	0h	Force Capture Event 3. This event is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CEVT3 flag.
2	CEVT2	R-0/W1S	0h	Force Capture Event 2. This event is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CEVT2 flag.

Table 28-18. ECFRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	CEVT1	R-0/W1S	0h	Force Capture Event 1. This event is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Sets the CEVT1 flag.
0	RESERVED	R	0h	Reserved

28.9.2.14 ECAPSYNCINSEL Register (Offset = 3Ch) [Reset = 00000001h]

ECAPSYNCINSEL is shown in Figure 28-32 and described in Table 28-19.

Return to the Summary Table.

SYNC source select register

Figure 28-32. ECAPSYNCINSEL Register

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

Table 28-19. ECAPSYNCINSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6-0	SEL	R/W	1h	These bits determines the source of SYNCIN signal. 0x0 : Disabled using SOC tieoff. 0x7F : Refer to SOC spec for details. Reset type: SYSRSn

28.9.3 ECAP_SIGNAL_MONITORING Registers

Table 28-20 lists the memory-mapped registers for the ECAP_SIGNAL_MONITORING registers. All register offset addresses not listed in Table 28-20 should be considered as reserved locations and the register contents should not be modified.

Table 28-20. ECAP_SIGNAL_MONITORING Registers

Offset	Acronym	Register Name	Protection
0h	MUNIT_COMMON_CTL	Control registers for monitoring unit {#}	
40h	MUNIT_1_CTL	Control registers for monitoring unit 1	
44h	MUNIT_1_SHADOW_CTL	Shadow control registers for monitoring unit 1	
50h	MUNIT_1_MIN	Min value for monitoring unit 1	
54h	MUNIT_1_MAX	Max value for monitoring unit 1	
58h	MUNIT_1_MIN_SHADOW	Shadow register for Min value of monitoring unit 1	
5Ch	MUNIT_1_MAX_SHADOW	Shadow register for Max value of monitoring unit 1	
60h	MUNIT_1_DEBUG_RANGE_MIN	Observed Min value of check being enabled on minotoring unit 1	
64h	MUNIT_1_DEBUG_RANGE_MAX	Observed Max value of check being enabled on minotoring unit 1	
80h	MUNIT_2_CTL	Control registers for monitoring unit 2	
84h	MUNIT_2_SHADOW_CTL	Shadow control registers for monitoring unit 2	
90h	MUNIT_2_MIN	Min value for monitoring unit 2	
94h	MUNIT_2_MAX	Max value for monitoring unit 2	
98h	MUNIT_2_MIN_SHADOW	Shadow register for Min value of monitoring unit 2	
9Ch	MUNIT_2_MAX_SHADOW	Shadow register for Max value of monitoring unit 2	
A0h	MUNIT_2_DEBUG_RANGE_MIN	Observed Min value of check being enabled on minotoring unit 2	
A4h	MUNIT_2_DEBUG_RANGE_MAX	Observed Max value of check being enabled on minotoring unit 2	

Complex bit access types are encoded to fit into small table cells. Table 28-21 shows the codes that are used for access types in this section.

Table 28-21. ECAP_SIGNAL_MONITORING Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.

Table 28-21. ECAP_SIGNAL_MONITORING Access Type Codes (continued)

Access Type	Code	Description
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

28.9.3.1 MUNIT_COMMON_CTL Register (Offset = 0h) [Reset = 00000000h]

MUNIT_COMMON_CTL is shown in Figure 28-33 and described in Table 28-22.

Return to the Summary Table.

Control registers for monitoring unit {#}

Figure 28-33. MUNIT_COMMON_CTL Register

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

Table 28-22. MUNIT_COMMON_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	RESERVED	R-0	0h	Reserved
14-8	GLDSTRBSEL	R/W	0h	Global load strobe select to enable shadow to active loading 0x0 : Disabled with SOC level tieoff. 0x1 to 0x7F : Global load strobe from SOC level including ETPWM global load strobes. Reset type: CPU1.SYSRSn
7	RESERVED	R-0	0h	Reserved
6-0	TRIPSEL	R/W	0h	Trip signal select to disable and enable signal monitoring automatically 0x0 : Disabled, Trip signals does not affect signal monitoring, achieved with SOC level tieoff. 0x1 to 0x7F : Signal monioring is disabled when selected signal is high and enabled when it is low Reset type: CPU1.SYSRSn

28.9.3.2 MUNIT_1_CTL Register (Offset = 40h) [Reset = 00000000h]

MUNIT_1_CTL is shown in Figure 28-34 and described in Table 28-23.

Return to the Summary Table.

Control registers for monitoring unit 1

Figure 28-34. MUNIT_1_CTL Register

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

Table 28-23. MUNIT_1_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-12	RESERVED	R-0	0h	Reserved
11-8	MON_SEL	R/W	0h	Type of monitoring 0 : High Pulse width 1 : Low Pulse width 2 : Period width from Rise to Rise 3 : Period width from fall to fall 4 : Monitor rise edge 5 : Monitor fall edge 6-15 : Reserved (High Pulse width) Reset type: CPU1.SYSRSn
7-3	RESERVED	R-0	0h	Reserved
2	DISABLE_EARLY_MAX_ERR	R/W	0h	Disable early max error check 0 : Max error is generated as soon as pulse width is greater than specified max value 1 : Max error is generated when second event has occurred Reset type: CPU1.SYSRSn
1	DEBUG_RANGE_EN	R/W	0h	Debug mode enable. 0 : Debug mode is disabled. 1 : Debug mode of monitoring unit 1 is enabled to obtain the variation seen in the system for debug purpose. Range is captured in MUNIT_1_DEBUG_RANGE_MIN and MUNIT_1_DEBUG_RANGE_MAX registers Toggle DEBUG_RANGE_EN to restart this process, this will initialize MUNIT_1_DEBUG_RANGE_MIN and MUNIT_1_DEBUG_RANGE_MAX registers. Reset type: CPU1.SYSRSn
0	EN	R/W	0h	0 : Monitoring unit 1 is disabled 1 : Monitoring unit 1 is enabled Reset type: CPU1.SYSRSn

28.9.3.3 MUNIT_1_SHADOW_CTL Register (Offset = 44h) [Reset = 00000000h]

MUNIT_1_SHADOW_CTL is shown in Figure 28-35 and described in Table 28-24.

Return to the Summary Table.

Shadow control registers for monitoring unit 1

Figure 28-35. MUNIT_1_SHADOW_CTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					LOADMODE	SWSYNC	SYNCI_EN
R-0-0h					R/W-0h	R-0/W1S-0h	R/W-0h

Table 28-24. MUNIT_1_SHADOW_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	LOADMODE	R/W	0h	Load mode 0 : Active registers are loaded with shadow on next sync event 1 : Active registers are loaded with shadow on EPWMx.GLDLCSTRB event Reset type: CPU1.SYSRSn
1	SWSYNC	R-0/W1S	0h	Copies Min and Max values from shadow to active registers immediately if MUNIT_1_SHADOW_CTL.SYNCI_EN is set. Reset type: CPU1.SYSRSn
0	SYNCI_EN	R/W	0h	Shadow Enable 0 : Disabled 1 : Enabled Reset type: CPU1.SYSRSn

28.9.3.4 MUNIT_1_MIN Register (Offset = 50h) [Reset = 00000000h]

MUNIT_1_MIN is shown in Figure 28-36 and described in Table 28-25.

Return to the Summary Table.

Min value for monitoring unit 1

Figure 28-36. MUNIT_1_MIN Register

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

Table 28-25. MUNIT_1_MIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MIN_VALUE	R/W	0h	Minimum value for monitoring Reset type: CPU1.SYSRSn

28.9.3.5 MUNIT_1_MAX Register (Offset = 54h) [Reset = 00000000h]

MUNIT_1_MAX is shown in Figure 28-37 and described in Table 28-26.

Return to the Summary Table.

Max value for monitoring unit 1

Figure 28-37. MUNIT_1_MAX Register

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

Table 28-26. MUNIT_1_MAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MAX_VALUE	R/W	0h	Maximum value for monitoring Reset type: CPU1.SYSRSn

28.9.3.6 MUNIT_1_MIN_SHADOW Register (Offset = 58h) [Reset = 00000000h]

MUNIT_1_MIN_SHADOW is shown in Figure 28-38 and described in Table 28-27.

Return to the Summary Table.

Shadow register for Min value of monitoring unit 1

Figure 28-38. MUNIT_1_MIN_SHADOW Register

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

Table 28-27. MUNIT_1_MIN_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MIN_VALUE_SHADOW	R/W	0h	Shadow minimum value for monitoring. Shadow value is loaded to active register on Sync event or global load strobe. Reset type: CPU1.SYSRSn

28.9.3.7 MUNIT_1_MAX_SHADOW Register (Offset = 5Ch) [Reset = 00000000h]

MUNIT_1_MAX_SHADOW is shown in Figure 28-39 and described in Table 28-28.

Return to the Summary Table.

Shadow register for Max value of monitoring unit 1

Figure 28-39. MUNIT_1_MAX_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_VALUE_SHADOW																															
R/W-0h																															

Table 28-28. MUNIT_1_MAX_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MAX_VALUE_SHADOW	R/W	0h	Shadow maximum value for monitoring. Shadow value is loaded to active register on Sync event or global load strobe. Reset type: CPU1.SYSRSn

28.9.3.8 MUNIT_1_DEBUG_RANGE_MIN Register (Offset = 60h) [Reset = FFFFFFFFh]

MUNIT_1_DEBUG_RANGE_MIN is shown in Figure 28-40 and described in Table 28-29.

Return to the Summary Table.

Observed Min value of check being enabled on minotoring unit 1

Figure 28-40. MUNIT_1_DEBUG_RANGE_MIN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIN_VALUE																															
R-FFFFFFFFh																															

Table 28-29. MUNIT_1_DEBUG_RANGE_MIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MIN_VALUE	R	FFFFFFFFh	Observed Min value of check being enabled on minotoring unit 1. Is updated when MUNIT_1_CTL.DEBUG_RANGE_EN is set to '1' Reset type: CPU1.SYSRSn

28.9.3.9 MUNIT_1_DEBUG_RANGE_MAX Register (Offset = 64h) [Reset = 00000000h]

MUNIT_1_DEBUG_RANGE_MAX is shown in Figure 28-41 and described in Table 28-30.

Return to the Summary Table.

Observed Max value of check being enabled on minotoring unit 1

Figure 28-41. MUNIT_1_DEBUG_RANGE_MAX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_VALUE																															
R-0h																															

Table 28-30. MUNIT_1_DEBUG_RANGE_MAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MAX_VALUE	R	0h	Observed Min value of check being enabled on minotoring unit 1. Is updated when MUNIT_1_CTL.DEBUG_RANGE_EN is set to '1' Reset type: CPU1.SYSRSn

28.9.3.10 MUNIT_2_CTL Register (Offset = 80h) [Reset = 00000000h]

MUNIT_2_CTL is shown in Figure 28-42 and described in Table 28-31.

Return to the Summary Table.

Control registers for monitoring unit 2

Figure 28-42. MUNIT_2_CTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				MON_SEL			
R-0-0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED					DISABLE_EARLY_MAX_ERR	DEBUG_RANGE_EN	EN
R-0-0h					R/W-0h	R/W-0h	R/W-0h

Table 28-31. MUNIT_2_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-12	RESERVED	R-0	0h	Reserved
11-8	MON_SEL	R/W	0h	Type of monitoring 0 : High Pulse width 1 : Low Pulse width 2 : Period width from Rise to Rise 3 : Period width from fall to fall 4 : Monitor rise edge 5 : Monitor fall edge 6-15 : Reserved (High Pulse width) Reset type: CPU1.SYSRSn
7-3	RESERVED	R-0	0h	Reserved
2	DISABLE_EARLY_MAX_ERR	R/W	0h	Disable early max error check 0 : Max error is generated as soon as pulse width is greater than specified max value 1 : Max error is generated when second event has occurred Reset type: CPU1.SYSRSn
1	DEBUG_RANGE_EN	R/W	0h	Debug mode enable. 0 : Debug mode is disabled. 1 : Debug mode of monitoring unit 2 is enabled to obtain the variation seen in the system for debug purpose. Range is captured in MUNIT_2_DEBUG_RANGE_MIN and MUNIT_2_DEBUG_RANGE_MAX registers Toggle DEBUG_RANGE_EN to restart this process, this will initialize MUNIT_2_DEBUG_RANGE_MIN and MUNIT_2_DEBUG_RANGE_MAX registers. Reset type: CPU1.SYSRSn
0	EN	R/W	0h	0 : Monitoring unit 2 is disabled 1 : Monitoring unit 2 is enabled Reset type: CPU1.SYSRSn

28.9.3.11 MUNIT_2_SHADOW_CTL Register (Offset = 84h) [Reset = 00000000h]

MUNIT_2_SHADOW_CTL is shown in Figure 28-43 and described in Table 28-32.

Return to the Summary Table.

Shadow control registers for monitoring unit 2

Figure 28-43. MUNIT_2_SHADOW_CTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					LOADMODE	SWSYNC	SYNCI_EN
R-0-0h					R/W-0h	R-0/W1S-0h	R/W-0h

Table 28-32. MUNIT_2_SHADOW_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	LOADMODE	R/W	0h	Load mode 0 : Active registers are loaded with shadow on next sync event 1 : Active registers are loaded with shadow on EPWMx.GLDLCSTRB event Reset type: CPU1.SYSRSn
1	SWSYNC	R-0/W1S	0h	Copies Min and Max values from shadow to active registers immediately if MUNIT_2_SHADOW_CTL.SYNCI_EN is set. Reset type: CPU1.SYSRSn
0	SYNCI_EN	R/W	0h	Shadow Enable 0 : Disabled 1 : Enabled Reset type: CPU1.SYSRSn

28.9.3.12 MUNIT_2_MIN Register (Offset = 90h) [Reset = 00000000h]

MUNIT_2_MIN is shown in Figure 28-44 and described in Table 28-33.

Return to the Summary Table.

Min value for monitoring unit 2

Figure 28-44. MUNIT_2_MIN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIN_VALUE																															
R/W-0h																															

Table 28-33. MUNIT_2_MIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MIN_VALUE	R/W	0h	Minimum value for monitoring Reset type: CPU1.SYSRSn

28.9.3.13 MUNIT_2_MAX Register (Offset = 94h) [Reset = 00000000h]

MUNIT_2_MAX is shown in Figure 28-45 and described in Table 28-34.

Return to the Summary Table.

Max value for monitoring unit 2

Figure 28-45. MUNIT_2_MAX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_VALUE																															
R/W-0h																															

Table 28-34. MUNIT_2_MAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MAX_VALUE	R/W	0h	Maximum value for monitoring Reset type: CPU1.SYSRSn

28.9.3.14 MUNIT_2_MIN_SHADOW Register (Offset = 98h) [Reset = 00000000h]

MUNIT_2_MIN_SHADOW is shown in Figure 28-46 and described in Table 28-35.

Return to the Summary Table.

Shadow register for Min value of monitoring unit 2

Figure 28-46. MUNIT_2_MIN_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIN_VALUE_SHADOW																															
R/W-0h																															

Table 28-35. MUNIT_2_MIN_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MIN_VALUE_SHADOW	R/W	0h	Shadow minimum value for monitoring. Shadow value is loaded to active register on Sync event or global load strobe. Reset type: CPU1.SYSRSn

28.9.3.15 MUNIT_2_MAX_SHADOW Register (Offset = 9Ch) [Reset = 00000000h]

MUNIT_2_MAX_SHADOW is shown in Figure 28-47 and described in Table 28-36.

Return to the Summary Table.

Shadow register for Max value of monitoring unit 2

Figure 28-47. MUNIT_2_MAX_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_VALUE_SHADOW																															
R/W-0h																															

Table 28-36. MUNIT_2_MAX_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MAX_VALUE_SHADOW	R/W	0h	Shadow maximum value for monitoring. Shadow value is loaded to active register on Sync event or global load strobe. Reset type: CPU1.SYSRSn

28.9.3.16 MUNIT_2_DEBUG_RANGE_MIN Register (Offset = A0h) [Reset = FFFFFFFFh]

MUNIT_2_DEBUG_RANGE_MIN is shown in Figure 28-48 and described in Table 28-37.

Return to the Summary Table.

Observed Min value of check being enabled on minotoring unit 2

Figure 28-48. MUNIT_2_DEBUG_RANGE_MIN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIN_VALUE																															
R-FFFFFFFFh																															

Table 28-37. MUNIT_2_DEBUG_RANGE_MIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MIN_VALUE	R	FFFFFFFFh	Observed Min value of check being enabled on minotoring unit 2. Is updated when MUNIT_2_CTL.DEBUG_RANGE_EN is set to '1' Reset type: CPU1.SYSRSn

28.9.3.17 MUNIT_2_DEBUG_RANGE_MAX Register (Offset = A4h) [Reset = 00000000h]

MUNIT_2_DEBUG_RANGE_MAX is shown in Figure 28-49 and described in Table 28-38.

Return to the Summary Table.

Observed Max value of check being enabled on minotoring unit 2

Figure 28-49. MUNIT_2_DEBUG_RANGE_MAX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_VALUE																															
R-0h																															

Table 28-38. MUNIT_2_DEBUG_RANGE_MAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MAX_VALUE	R	0h	Observed Min value of check being enabled on minotoring unit 2. Is updated when MUNIT_2_CTL.DEBUG_RANGE_EN is set to '1' Reset type: CPU1.SYSRSn

28.9.4 HRCAP_REGS Registers

Table 28-39 lists the memory-mapped registers for the HRCAP_REGS registers. All register offset addresses not listed in Table 28-39 should be considered as reserved locations and the register contents should not be modified.

Table 28-39. HRCAP_REGS Registers

Offset	Acronym	Register Name	Protection
0h	HRCTL	High-Res Control Register	
8h	HRINTEN	High-Res Calibration Interrupt Enable Register	
Ch	HRFLG	High-Res Calibration Interrupt Flag Register	
10h	HRCLR	High-Res Calibration Interrupt Clear Register	
14h	HRFRC	High-Res Calibration Interrupt Force Register	
18h	HRCALPRD	High-Res Calibration Period Register	
1Ch	HRSYSCLKCTR	High-Res Calibration SYSCLK Counter Register	
20h	HRSYSCLKCAP	High-Res Calibration SYSCLK Capture Register	
24h	HRCLKCTR	High-Res Calibration HRCLK Counter Register	
28h	HRCLKCAP	High-Res Calibration HRCLK Capture Register	

Complex bit access types are encoded to fit into small table cells. Table 28-40 shows the codes that are used for access types in this section.

Table 28-40. HRCAP_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

28.9.4.1 HRCTL Register (Offset = 0h) [Reset = 00000000h]

HRCTL is shown in Figure 28-50 and described in Table 28-41.

Return to the Summary Table.

High-Res Control Register

Figure 28-50. HRCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED		CALIBCONT	CALIBSTS	CALIBSTART	PRDSEL	HRCLKE	HRE
R/W-0h		R/W-0h	R-0h	R-0/W1S-0h	R/W-0h	R/W-0h	R/W-0h

Table 28-41. HRCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	Reserved
5	CALIBCONT	R/W	0h	Continuous mode Calibration Select Bit: 0 Continuous mode disabled. 1 Continuous mode enabled. Calibration automatically restarts at end of current calibration cycle. Reset type: CPU1.SYSRSn
4	CALIBSTS	R	0h	Calibration status Bit: 0 No active calibration cycle 1 Calibration cycle in progress Reset type: CPU1.SYSRSn
3	CALIBSTART	R-0/W1S	0h	Calibration start Bit: 0 No effect 1 Starts the calibration cycle Reset type: CPU1.SYSRSn
2	PRDSEL	R/W	0h	Calibration Period Match Select Bit: 0 Use SYSCLK Counter For Period Match (default at reset) 1 Reserved Reset type: CPU1.SYSRSn
1	HRCLKE	R/W	0h	High Resolution Clock Enable Bit: 0 High resolution clock disabled (default at reset) 1 High resolution clock enabled. The clock should be enabled before enabling the high res function via the HRE bit. Reset type: CPU1.SYSRSn

Table 28-41. HRCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	HRE	R/W	0h	High Resolution Enable Bit: 0 High resolution mode disabled (default at reset) 1 High resolution mode enabled. Enabling this mode will connect the capture registers and edge event modes of the ECAP to be accessed by the High Res function. Note: The High Res clock needs to be enabled (using the HRCLKE bit) first before enabling the module. Allow a certain start up stabilization period before enabling the module. Reset type: CPU1.SYSRSn

28.9.4.2 HRINTEN Register (Offset = 8h) [Reset = 00000000h]

HRINTEN is shown in Figure 28-51 and described in Table 28-42.

Return to the Summary Table.

High-Res Calibration Interrupt Enable Register

Figure 28-51. HRINTEN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					CALPRDCHKSTS	CALIBDONE	RESERVED
R-0-0h					R/W-0h	R/W-0h	R-0-0h

Table 28-42. HRINTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	CALPRDCHKSTS	R/W	0h	Calibration Period Check status Interrupt Enable: 0 Disable Calibration Period Check interrupt status 1 Enable Calibration Period Check interrupt status Reset type: CPU1.SYSRSn
1	CALIBDONE	R/W	0h	Calibration done Interrupt Enable: 0 Disable Calibration done Interrupt 1 Enable Calibration done Interrupt Reset type: CPU1.SYSRSn
0	RESERVED	R-0	0h	Reserved

28.9.4.3 HRFLG Register (Offset = Ch) [Reset = 00000000h]

HRFLG is shown in Figure 28-52 and described in Table 28-43.

Return to the Summary Table.

High-Res Calibration Interrupt Flag Register

Figure 28-52. HRFLG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					CALPRDCHKSTS	CALIBDONE	CALIBINT
R-0-0h					R-0h	R-0h	R-0h

Table 28-43. HRFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	CALPRDCHKSTS	R	0h	Calibration period check status Flag Bit: 1 Indicates that calibration ended before PRDCHK due to overflow on one of the counters. 0 Indicates no event occurred. Note: This bit remains latched until cleared by the user using the HRCLR [CALPRDCHKSTS] bit. Reset type: CPU1.SYSRSn
1	CALIBDONE	R	0h	Calibration Done Interrupt Flag Bit: 1 Indicates calibration cycle is completed 0 Indicates calibration cycle has not completed. Note: This bit remains latched until cleared by the user using the HRCLR [CALIBDONE] bit. Reset type: CPU1.SYSRSn
0	CALIBINT	R	0h	Global calibration Interrupt Status Flag: 1 Indicates that an interrupt was generated from CALIBDONE or CALPRDCHKSTS. 0 Indicates no interrupt generated. Reset type: CPU1.SYSRSn

28.9.4.4 HRCLR Register (Offset = 10h) [Reset = 00000000h]

HRCLR is shown in Figure 28-53 and described in Table 28-44.

Return to the Summary Table.

High-Res Calibration Interrupt Clear Register

Figure 28-53. HRCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					CALPRDCHKSTS	CALIBDONE	CALIBINT
R-0-0h					R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 28-44. HRCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	CALPRDCHKSTS	R-0/W1C	0h	Clear Calibration period check status Flag Bit: 1 Clears the CALPRDCHKSTS flag register bit. 0 No effect. Note: H/W has priority over CPU writes if the user tries to clear a flag bit and an event occurs on the same cycle that tries to set the flag for the selected bit. Reset type: CPU1.SYSRSn
1	CALIBDONE	R-0/W1C	0h	Clear Calibration Done Interrupt Flag Bit: 1 Clears the CALIBDONE interrupt flag register bit. 0 No effect. Note: H/W has priority over CPU writes if the user tries to clear a flag bit and an event occurs on the same cycle that tries to set the flag for the selected bit. Reset type: CPU1.SYSRSn
0	CALIBINT	R-0/W1C	0h	Clear Global calibration Interrupt Flag 1 Clears the Global interrupt flag and enables further interrupts to be generated if any of the event flags are set. 0 No effect. Reset type: CPU1.SYSRSn

28.9.4.5 HRFRC Register (Offset = 14h) [Reset = 00000000h]

HRFRC is shown in Figure 28-54 and described in Table 28-45.

Return to the Summary Table.

High-Res Calibration Interrupt Force Register

Figure 28-54. HRFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					CALPRDCHKSTS	CALIBDONE	RESERVED
R-0-0h					R-0/W1S-0h	R-0/W1S-0h	R-0-0h

Table 28-45. HRFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	CALPRDCHKSTS	R-0/W1S	0h	Force CALPRDCHKSTS flag: 0 No effect 1 Sets the CALPRDCHKSTS flag. Reset type: CPU1.SYSRSn
1	CALIBDONE	R-0/W1S	0h	Force CALIBDONE flag: 0 No effect 1 Sets the CALIBDONE flag. Reset type: CPU1.SYSRSn
0	RESERVED	R-0	0h	Reserved

28.9.4.6 HRCALPRD Register (Offset = 18h) [Reset = 003FFFFFh]

HRCALPRD is shown in Figure 28-55 and described in Table 28-46.

Return to the Summary Table.

High-Res Calibration Period Register

Figure 28-55. HRCALPRD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRD																															
R/W-003FFFFFh																															

Table 28-46. HRCALPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PRD	R/W	003FFFFFh	Register to program calibration period. The period value is matched against HRSYSCLKCTR. On a match an interrupt is generated and the counter registers values are captured. Reset type: CPU1.SYSRSn

28.9.4.7 HRSYSCLKCTR Register (Offset = 1Ch) [Reset = 00000000h]

HRSYSCLKCTR is shown in Figure 28-56 and described in Table 28-47.

Return to the Summary Table.

High-Res Calibration SYSCLK Counter Register

Figure 28-56. HRSYSCLKCTR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HRSYSCLKCTR																															
R-0h																															

Table 28-47. HRSYSCLKCTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HRSYSCLKCTR	R	0h	Current SYSCLK counter value Reset type: CPU1.SYSRSn

28.9.4.8 HRSYSCLKCAP Register (Offset = 20h) [Reset = 00000000h]

HRSYSCLKCAP is shown in Figure 28-57 and described in Table 28-48.

Return to the Summary Table.

High-Res Calibration SYSCLK Capture Register

Figure 28-57. HRSYSCLKCAP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HRSYSCLKCAP																															
R-0h																															

Table 28-48. HRSYSCLKCAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HRSYSCLKCAP	R	0h	HRSYSCLKCTR is captures into this register at end of calibration cycle. Reset type: CPU1.SYSRSn

28.9.4.9 HRCLKCTR Register (Offset = 24h) [Reset = 00000000h]

HRCLKCTR is shown in Figure 28-58 and described in Table 28-49.

Return to the Summary Table.

High-Res Calibration HRCLK Counter Register

Figure 28-58. HRCLKCTR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HRCLKCTR																															
R-0h																															

Table 28-49. HRCLKCTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HRCLKCTR	R	0h	Current HRCLK counter value Note: HRCLK is not synchronized to SYSCLK domain so reads may not be accurate Reset type: CPU1.SYSRSn

28.9.4.10 HRCLKCAP Register (Offset = 28h) [Reset = 00000000h]

HRCLKCAP is shown in Figure 28-59 and described in Table 28-50.

Return to the Summary Table.

High-Res Calibration HRCLK Capture Register

Figure 28-59. HRCLKCAP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HRCLKCAP																															
R-0h																															

Table 28-50. HRCLKCAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HRCLKCAP	R	0h	HRCLKCTR is captures into this register at end of calibration cycle. Note: HRCLK is not synchronized to SYSCLK domain so reads may not be accurate Reset type: CPU1.SYSRSn

Chapter 29

High Resolution Capture (HRCAP)

This chapter describes the operation of the high resolution capture (HRCAP) module. The HRCAP submodule described here is part of the Type1 eCAP. HRCAP measures the width of external pulses to a higher degree of accuracy than the eCAP module. See the *C2000 Real-Time Control Peripheral Reference Guide* for a list of all devices with an HRCAP module of the same type, to determine the differences between types, and for a list of device-specific differences within a type. A detailed description of all referenced functions can be found in the C2000Ware documentation.

29.1 Introduction

Uses for the HRCAP module include:

- Capacitive touch applications
- High-resolution period and duty cycle measurements of pulse train cycles
- Instantaneous speed measurements
- Instantaneous frequency measurements
- Voltage measurements across an isolation boundary
- Distance/sonar measurement and scanning
- Measuring flow

29.1.1 HRCAP Related Collateral

Foundational Materials

- C28x Academy - HRCAP

Getting Started Materials

- *Leveraging High Resolution Capture (HRCAP) for Single Wire Data Transfer Application Report*

29.1.2 Features

The HRCAP module includes the following features:

- Pulse-width capture in either non-high-resolution or high-resolution modes
- Absolute mode pulse-width capture
- Continuous or one-shot capture
- Interrupt on either falling or rising edge
- Continuous mode capture of pulse widths in 4-deep buffer
- Hardware calibration logic for precision high-resolution capture

All of the previous resources are available on any pin using the Input X-BAR.

29.1.3 Description

Improvements from the Type 0 HRCAP are:

- Simplified calibration scheme:
  - HRCAP is always functional; never offline to perform calibration
  - Calibration is always running in the background; drastically reduced software overhead to calibrate
- Reduced software overhead to compute fractional bits
- Fractional and integer portions are packed into 32 bits
- All eCAP hardware is accessible when using the HRCAP enhancements. See Section 29.3 for practical considerations.
- Usage of the HRCAP is now unified with the eCAP

The HRCAP enhancement has been added to eCAP 5 and eCAP 6 to allow signals to be captured asynchronously to SYSCLK. Each HRCAP submodule includes one capture channel in addition to a hardware calibration block. All eCAP hardware is accessible when using the HRCAP enhancements; however, using the Event Filter or the Input Qualifier is not valid, as these are synchronous to SYSCLK.

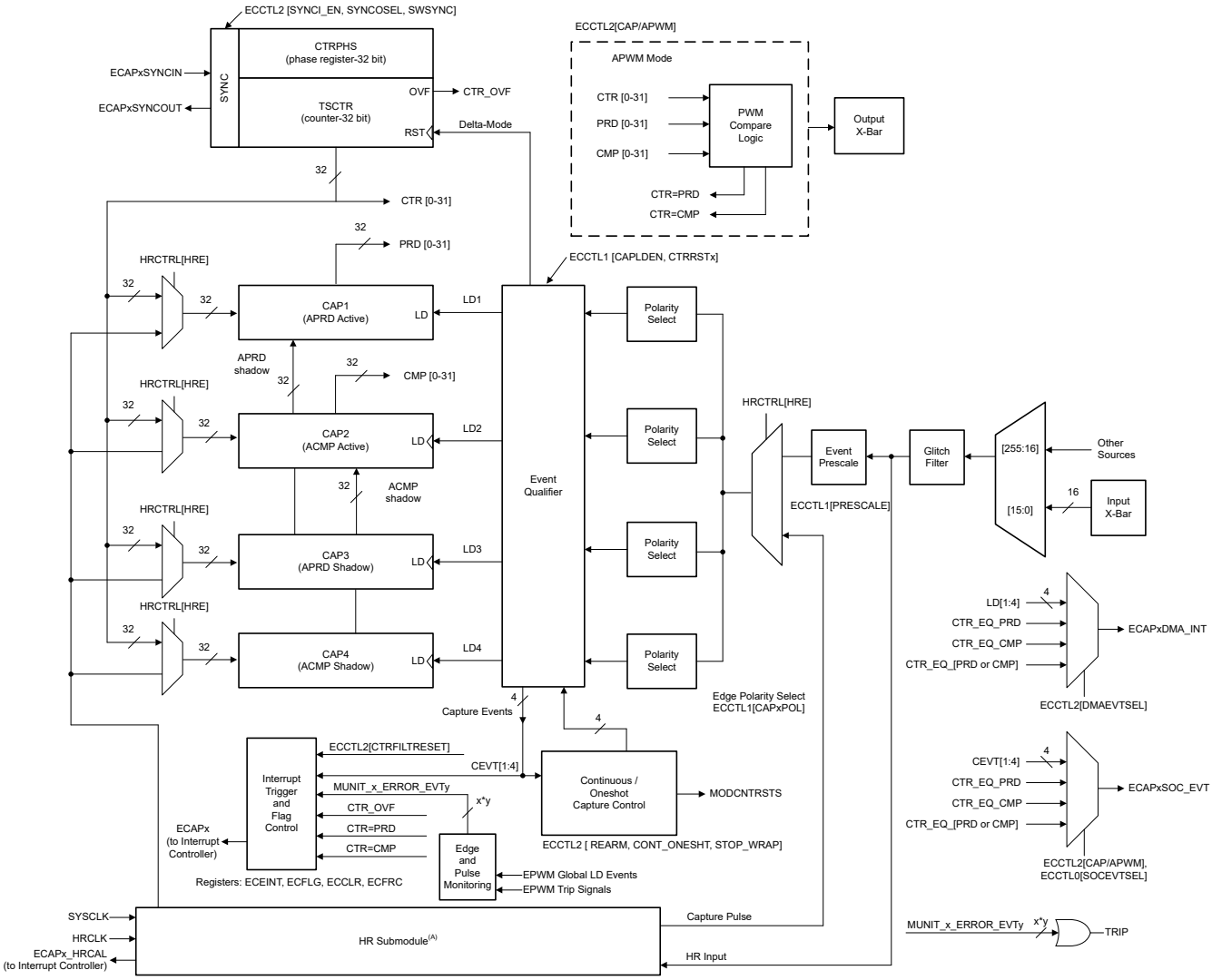
Each HRCAP-capable channel has the following independent key resources:

- All hardware of the respective eCAP
- High-resolution calibration logic
- Dedicated calibration interrupt

29.2 Operational Details

Figure 29-1 shows the various components that implement the high-resolution capture functionality of the eCAP module. Note that existing eCAP resources are reused, which requires that the eCAP module is set up before using the HRCAP enhancements. For simplicity, absolute timestamp measurements are recommended. See Section 29.3 for more details.

All HRCAP measurements are relative-time measures, in terms of minimum step size. Calibration hardware as well as software functions, have been provided to convert relative-time measurements to time-converted measurements in terms of seconds. The calibration hardware and software is only required if time-converted measurements are required.



A. The HRCAP submodule is not available on all eCAP modules; in this case, the high-resolution muxes and hardware are not implemented.

Figure 29-1. HRCAP Operations Block Diagram

29.2.1 HRCAP Clocking

Unlike previous Type-0 HRCAP modules, the Type-1 eCAP, with HRCAP functionality, does not require a second PLL. However, the module still requires both SYSCLK and a second asynchronous clock source called HRCLK. The HRCLK is sensitive to changes in both temperature and voltage. For this reason, when using time-converted measurements, it is required to make periodic continuous calibrations.

29.2.2 HRCAP Initialization Sequence

Following are the HRCAP initialization sequence steps. When using the HRCAP to take relative-time measurements, only steps 1-5 are required. When using the HRCAP to take time-converted measurements, all steps are required.

1. Enable HRCLK using HRCAP_enableHighResolutionClock()
2. Delay 1µs
3. Enable HR mode using HRCAP_enableHighResolution()
4. Delay 1µs
5. Configure the eCAP module as desired, including interrupts
6. Set calibration period using HRCAP_setCalibrationPeriod()
7. Enable continuous calibration using HRCAP_setCalibrationMode()
8. Enable interrupts using HRCAP_enableCalibrationInterrupt()
9. Start calibration using HRCAP_startCalibration()

29.2.3 HRCAP Interrupts

The HRCAP enhancements leverage the existing eCAP interrupts (see *Interrupt Control* in the ECAP chapter) in addition to HRCALINT, which is used exclusively by the hardware calibration block. HRCALINT can be triggered by the following conditions:

1. SYSCLKCTR = HRCALIBPERIOD
2. SYSCLKCTR or HRCLKCTR experience an overflow condition

Figure 29-2 shows this logic.

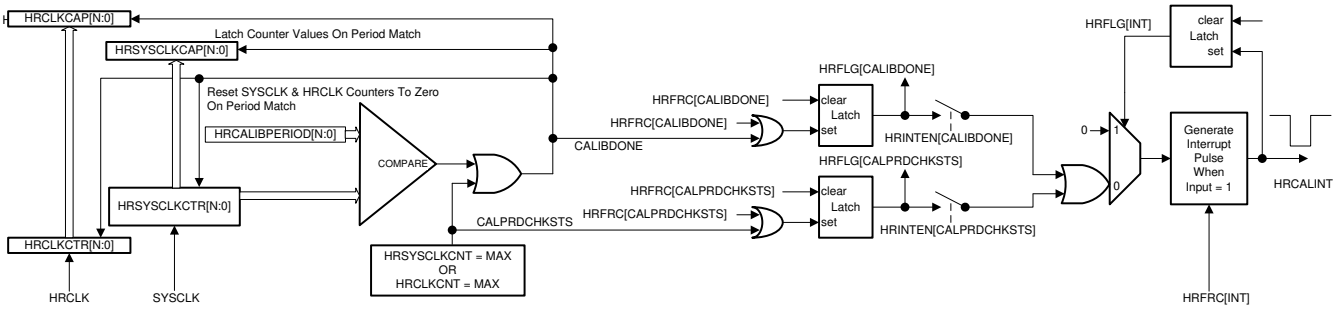


Figure 29-2. HRCAP Calibration

29.2.4 HRCAP Calibration

Note

For HRCAP calibration to work, the system needs to operate at SYSCLK frequency > 100MHz.

The following section applies only to time-converted measurements; calibration for relative-time measurements is not required. All values captured by the HRCAP submodule are in number of HRCLK cycles. The HRCLK speed varies widely with temperature and voltage, thus a scale factor is required to convert the capture value to the SYSCLK domain. For the same reason, periodically recalculate the scale factor. The HRCAP submodule has a calibration block to reduce software overhead when calculating a scale factor between HRCLK and SYSCLK.

The calibration block contains the following key resources:

- **HRSYSCLKCNT:** A 32-bit counter connected to SYSCLK. The counter starts counting when CALIBSTART is set.
- **HRCLKCNT:** A 32-bit counter connected to HRCLK. The counter starts counting when CALIBSTART is set.
- **HRCALIBPERIOD:** Calibration period, calibration is stopped when HRSYSCLKCNT is equal to the value in this register.
- **HRSYSCLKCAP:** On a calibration period match, the value of HRSYSCLKCNT is captured into HRSYSCLKCAP.
- **HRCLKCAP:** On a calibration period match, the value of HRCLKCNT is captured into HRCLKCAP.
- **HRCALINT:** An interrupt that occurs on a calibration period match, or when one of the counter registers experiences an overflow condition.

The calibration logic consists of two free-running counters; one clocked by HRCLK(HRCLKCTR) and one clocked by SYSCLK(HRSYSCLKCTR). When HRSYSCLKCTR is equal to HRCALIBPERIOD, the calibration block captures and resets both counter values, then triggers an interrupt, indicating a new scale factor is ready to be calculated. The scale factor can be found by dividing HRSYSCLKCAP by HRCLKCAP (see Equation 22). A DriverLib function, HRCAP_getScaleFactor, has been provided to determine the scale factor. This function can be called inside of the calibration interrupt service routine. If one of the counters experiences overflow, the CALPRDCHKSTS flag is set. The full details of the calibration block are described in Figure 29-2.

$$ScaleFactor = \frac{HRSYSCLKCAP}{HRCLKCAP} \quad (22)$$

Note

- Even with calibration, noise on the 1.2V VDD supply negatively affects the standard deviation of the HRCAP submodule. Care can be taken to make sure that the 1.2V supply is clean, and that noisy internal events such as enabling and disabling clock trees have been minimized while using the HRCAP submodule.
- When HRCLK > SYSCLK, calibration stops immaturely to mitigate improper calibration, SYSCLK must be set to at least 100MHz.

29.2.4.1 Applying the Scale Factor

A DriverLib function has been provided to apply the scale factor to a capture value, HRCAP_getEventTimeStampNanoseconds(). Equation 23 shows how to convert a raw count to seconds without using the DriverLib function.

$$Measurement(ns) = \frac{RawCount \times scaleFactor}{128} * SysClkPrd(ns) \quad (23)$$

Table 29-1. Scale Factor

Parameter	Explanation
RawCount	Capture value as read from ECAP_REGS_CAP1-4
ScaleFactor(1)	The Scale factor as calculated from Equation 23
128	Constant determined by the hardware of the HRCAP submodule
SysClkPrd(nS)	Period of the system clock
Measurement(nS)	Signal converted to nS

(1) The scale factor is not automatically applied to captured values. The user is required to apply the scale factor to all captured values as shown in Equation 23.

29.3 Known Exceptions

In HRCAP mode:

- Enabling and disabling core clocks negatively affects the standard deviation of the HRCAP submodule. Do not enable or disable core clocks while taking measurements.
- TSCTR is not writable; however, TSCTR can be reset using ECCTL2[CTRFILTRESET]
- Input synchronization is not applicable when using the HRCAP enhancements, because the HRCAP submodule is asynchronous to SYSCLK.
- The Event Filter functionality is not applicable for HRCAP, which defeats the purpose of HRCAP as the Event Filter's output is synchronous to SYSCLK.
- The best practice is to use absolute time mode for high-resolution mode. If time difference mode is used, it can lead to inaccurate results if the fractional value is not taken into consideration for capture events that have reset the time base counter.
  - Actual Capture Value = (Capture Value) – (fractional value of reference event that reset the counter)
- For high-frequency input signals, the CPU can not be able to cope with the speed of the captures. In such a case, one-shot mode is recommended. This mode allows the device to capture up to four edges before waiting to be serviced when the CPU is ready. This is applicable for the eCAP as well; however, in that case the event filter can be used to reduce the rate of captures.

29.4 Software

29.4.1 HRCAP Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/hrcap

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

29.4.1.1 HRCAP Capture and Calibration Example - SINGLE_CORE

FILE: hrcap_ex1_capture.c

This example configures an ECAP to use HRCAP functionality to capture time between edges on input GPIO2.

External Connections
The user must provide a signal to GPIO2. XCLKOUT has been configured to an output GPIO and can be externally jumped to serve this purpose. See Sysconfig file for XCLKOUT GPIO selected.

Watch Variables
- onTime1, onTime2
- offTime1, offTime2
- period1, period2

29.5 HRCAP Registers

This Section describes the HRCAP Registers.

29.5.1 HRCAP Base Address Table

Table 29-2. HRCAP Base Address Table

Bit Field Name		Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM-M4	Pipeline Protected
Structure	DriverLib Name									
HRCAP_REGS	HRCAP5_BASE	0x7010_4040	YES	YES	YES	YES	YES	YES	-	YES
HRCAP_REGS	HRCAP6_BASE	0x7010_5040	YES	YES	YES	YES	YES	YES	-	YES

29.5.2 HRCAP_REGS Registers

Table 29-3 lists the memory-mapped registers for the HRCAP_REGS registers. All register offset addresses not listed in Table 29-3 should be considered as reserved locations and the register contents should not be modified.

Table 29-3. HRCAP_REGS Registers

Offset	Acronym	Register Name	Protection
0h	HRCTL	High-Res Control Register	
8h	HRINTEN	High-Res Calibration Interrupt Enable Register	
Ch	HRFLG	High-Res Calibration Interrupt Flag Register	
10h	HRCLR	High-Res Calibration Interrupt Clear Register	
14h	HRFRC	High-Res Calibration Interrupt Force Register	
18h	HRCALPRD	High-Res Calibration Period Register	
1Ch	HRSYSCLKCTR	High-Res Calibration SYSCLK Counter Register	
20h	HRSYSCLKCAP	High-Res Calibration SYSCLK Capture Register	
24h	HRCLKCTR	High-Res Calibration HRCLK Counter Register	
28h	HRCLKCAP	High-Res Calibration HRCLK Capture Register	

Complex bit access types are encoded to fit into small table cells. Table 29-4 shows the codes that are used for access types in this section.

Table 29-4. HRCAP_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

29.5.2.1 HRCTL Register (Offset = 0h) [Reset = 00000000h]

HRCTL is shown in Figure 29-3 and described in Table 29-5.

Return to the Summary Table.

High-Res Control Register

Figure 29-3. HRCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED		CALIBCONT	CALIBSTS	CALIBSTART	PRDSEL	HRCLKE	HRE
R/W-0h		R/W-0h	R-0h	R-0/W1S-0h	R/W-0h	R/W-0h	R/W-0h

Table 29-5. HRCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	Reserved
5	CALIBCONT	R/W	0h	Continuous mode Calibration Select Bit: 0 Continuous mode disabled. 1 Continuous mode enabled. Calibration automatically restarts at end of current calibration cycle. Reset type: CPU1.SYSRSn
4	CALIBSTS	R	0h	Calibration status Bit: 0 No active calibration cycle 1 Calibration cycle in progress Reset type: CPU1.SYSRSn
3	CALIBSTART	R-0/W1S	0h	Calibration start Bit: 0 No effect 1 Starts the calibration cycle Reset type: CPU1.SYSRSn
2	PRDSEL	R/W	0h	Calibration Period Match Select Bit: 0 Use SYSCLK Counter For Period Match (default at reset) 1 Reserved Reset type: CPU1.SYSRSn
1	HRCLKE	R/W	0h	High Resolution Clock Enable Bit: 0 High resolution clock disabled (default at reset) 1 High resolution clock enabled. The clock should be enabled before enabling the high res function via the HRE bit. Reset type: CPU1.SYSRSn

Table 29-5. HRCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	HRE	R/W	0h	High Resolution Enable Bit: 0 High resolution mode disabled (default at reset) 1 High resolution mode enabled. Enabling this mode will connect the capture registers and edge event modes of the ECAP to be accessed by the High Res function. Note: The High Res clock needs to be enabled (using the HRCLKE bit) first before enabling the module. Allow a certain start up stabilization period before enabling the module. Reset type: CPU1.SYSRSn

29.5.2.2 HRINTEN Register (Offset = 8h) [Reset = 00000000h]

HRINTEN is shown in Figure 29-4 and described in Table 29-6.

Return to the Summary Table.

High-Res Calibration Interrupt Enable Register

Figure 29-4. HRINTEN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					CALPRDCHKSTS	CALIBDONE	RESERVED
R-0-0h					R/W-0h	R/W-0h	R-0-0h

Table 29-6. HRINTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	CALPRDCHKSTS	R/W	0h	Calibration Period Check status Interrupt Enable: 0 Disable Calibration Period Check interrupt status 1 Enable Calibration Period Check interrupt status Reset type: CPU1.SYSRSn
1	CALIBDONE	R/W	0h	Calibration done Interrupt Enable: 0 Disable Calibration done Interrupt 1 Enable Calibration done Interrupt Reset type: CPU1.SYSRSn
0	RESERVED	R-0	0h	Reserved

29.5.2.3 HRFLG Register (Offset = Ch) [Reset = 00000000h]

HRFLG is shown in Figure 29-5 and described in Table 29-7.

Return to the Summary Table.

High-Res Calibration Interrupt Flag Register

Figure 29-5. HRFLG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					CALPRDCHKSTS	CALIBDONE	CALIBINT
R-0-0h					R-0h	R-0h	R-0h

Table 29-7. HRFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	CALPRDCHKSTS	R	0h	Calibration period check status Flag Bit: 1 Indicates that calibration ended before PRDCHK due to overflow on one of the counters. 0 Indicates no event occurred. Note: This bit remains latched until cleared by the user using the HRCLR [CALPRDCHKSTS] bit. Reset type: CPU1.SYSRSn
1	CALIBDONE	R	0h	Calibration Done Interrupt Flag Bit: 1 Indicates calibration cycle is completed 0 Indicates calibration cycle has not completed. Note: This bit remains latched until cleared by the user using the HRCLR [CALIBDONE] bit. Reset type: CPU1.SYSRSn
0	CALIBINT	R	0h	Global calibration Interrupt Status Flag: 1 Indicates that an interrupt was generated from CALIBDONE or CALPRDCHKSTS. 0 Indicates no interrupt generated. Reset type: CPU1.SYSRSn

29.5.2.4 HRCLR Register (Offset = 10h) [Reset = 00000000h]

HRCLR is shown in Figure 29-6 and described in Table 29-8.

Return to the Summary Table.

High-Res Calibration Interrupt Clear Register

Figure 29-6. HRCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					CALPRDCHKSTS	CALIBDONE	CALIBINT
R-0-0h					R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 29-8. HRCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	CALPRDCHKSTS	R-0/W1C	0h	Clear Calibration period check status Flag Bit: 1 Clears the CALPRDCHKSTS flag register bit. 0 No effect. Note: H/W has priority over CPU writes if the user tries to clear a flag bit and an event occurs on the same cycle that tries to set the flag for the selected bit. Reset type: CPU1.SYSRSn
1	CALIBDONE	R-0/W1C	0h	Clear Calibration Done Interrupt Flag Bit: 1 Clears the CALIBDONE interrupt flag register bit. 0 No effect. Note: H/W has priority over CPU writes if the user tries to clear a flag bit and an event occurs on the same cycle that tries to set the flag for the selected bit. Reset type: CPU1.SYSRSn
0	CALIBINT	R-0/W1C	0h	Clear Global calibration Interrupt Flag 1 Clears the Global interrupt flag and enables further interrupts to be generated if any of the event flags are set. 0 No effect. Reset type: CPU1.SYSRSn

29.5.2.5 HRFRC Register (Offset = 14h) [Reset = 00000000h]

HRFRC is shown in Figure 29-7 and described in Table 29-9.

Return to the Summary Table.

High-Res Calibration Interrupt Force Register

Figure 29-7. HRFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					CALPRDCHKSTS	CALIBDONE	RESERVED
R-0-0h					R-0/W1S-0h	R-0/W1S-0h	R-0-0h

Table 29-9. HRFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	CALPRDCHKSTS	R-0/W1S	0h	Force CALPRDCHKSTS flag: 0 No effect 1 Sets the CALPRDCHKSTS flag. Reset type: CPU1.SYSRSn
1	CALIBDONE	R-0/W1S	0h	Force CALIBDONE flag: 0 No effect 1 Sets the CALIBDONE flag. Reset type: CPU1.SYSRSn
0	RESERVED	R-0	0h	Reserved

29.5.2.6 HRCALPRD Register (Offset = 18h) [Reset = 003FFFFFh]

HRCALPRD is shown in Figure 29-8 and described in Table 29-10.

Return to the Summary Table.

High-Res Calibration Period Register

Figure 29-8. HRCALPRD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRD																															
R/W-003FFFFFh																															

Table 29-10. HRCALPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PRD	R/W	003FFFFFh	Register to program calibration period. The period value is matched against HRSYSCLKCTR. On a match an interrupt is generated and the counter registers values are captured. Reset type: CPU1.SYSRSn

29.5.2.7 HRSYSCLKCTR Register (Offset = 1Ch) [Reset = 00000000h]

HRSYSCLKCTR is shown in Figure 29-9 and described in Table 29-11.

Return to the Summary Table.

High-Res Calibration SYSCLK Counter Register

Figure 29-9. HRSYSCLKCTR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HRSYSCLKCTR																															
R-0h																															

Table 29-11. HRSYSCLKCTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HRSYSCLKCTR	R	0h	Current SYSCLK counter value Reset type: CPU1.SYSRSn

29.5.2.8 HRSYSCLKCAP Register (Offset = 20h) [Reset = 00000000h]

HRSYSCLKCAP is shown in Figure 29-10 and described in Table 29-12.

Return to the Summary Table.

High-Res Calibration SYSCLK Capture Register

Figure 29-10. HRSYSCLKCAP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HRSYSCLKCAP																															
R-0h																															

Table 29-12. HRSYSCLKCAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HRSYSCLKCAP	R	0h	HRSYSCLKCTR is captures into this register at end of calibration cycle. Reset type: CPU1.SYSRSn

29.5.2.9 HRCLKCTR Register (Offset = 24h) [Reset = 00000000h]

HRCLKCTR is shown in Figure 29-11 and described in Table 29-13.

Return to the Summary Table.

High-Res Calibration HRCLK Counter Register

Figure 29-11. HRCLKCTR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HRCLKCTR																															
R-0h																															

Table 29-13. HRCLKCTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HRCLKCTR	R	0h	Current HRCLK counter value Note: HRCLK is not synchronized to SYSCLK domain so reads may not be accurate Reset type: CPU1.SYSRSn

29.5.2.10 HRCLKCAP Register (Offset = 28h) [Reset = 00000000h]

HRCLKCAP is shown in Figure 29-12 and described in Table 29-14.

Return to the Summary Table.

High-Res Calibration HRCLK Capture Register

Figure 29-12. HRCLKCAP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HRCLKCAP																															
R-0h																															

Table 29-14. HRCLKCAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HRCLKCAP	R	0h	HRCLKCTR is captures into this register at end of calibration cycle. Note: HRCLK is not synchronized to SYSCLK domain so reads may not be accurate Reset type: CPU1.SYSRSn

Chapter 30

Enhanced Pulse Width Modulator (ePWM)

The enhanced pulse width modulator (ePWM) peripheral is a key element in controlling many of the power electronic systems found in both commercial and industrial equipment. These systems include digital motor control, switch mode power supply control, uninterruptible power supplies (UPS), and other forms of power conversion. The ePWM peripheral can also perform a digital-to-analog (DAC) function, where the duty cycle is equivalent to a DAC analog value; it is sometimes referred to as a power DAC.

This chapter is applicable for ePWM type 5. Type 5 EPWM is fully compatible with type 4 EPWM. See the *C2000 Real-Time Control Peripheral Reference Guide* for a list of all devices with an ePWM module of the same type, to determine the differences between the types, and for a list of device-specific differences within a type.

30.1 Introduction

This chapter includes an overview and information about each submodule:

- Time Base (TB) Submodule
- Counter Compare (CC) Submodule
- Action Qualifier (AQ) Submodule
- Dead-Band Generator (DB) Submodule
- PWM Chopper (PC) Submodule
- Trip Zone (TZ) Submodule
- Diode Emulation (DE) Submodule
- Minimum Dead-Band (MINDB) and Illegal Combo Logic (ICL) Submodule
- Event Trigger (ET) Submodule
- Digital Compare (DC) Submodule

The ePWM Type 5 is functionally compatible to Type 4. Type 5 has the following enhancements in addition to the Type 4 features:

- **PWM SYNC Related Enhancements:** Additional external sync option is added in to the EPWMSYNCSEL register. This allows for the configuration of up to 3 independent sync chains with external sync options.
- **Linking and Global Load Enhancements:** DBRED:DBREDHR and DBREDHR and DBFED:DBFEDHR have the ability to be linked across ePWM modules.

  Global load pulse selection for shadow to active load can now occur when the time-base counter equals CMPCU, CMPCD, CMPDU, or CMPDD.
- **XCMP Complex Waveform Generator:** XCMP mode has been added to allow for generation of multiple ePWM pulses, with high resolution, in a given ePWM cycle. Up to 8 new compare registers are added to achieve this functionality.
- **Digital Compare Submodule Enhancements:** Event detection within the digital compare capture module is able to detect an occurrence of a trip event in a configured time window.

  Pulse selection for blanking and capture alignment now includes a blanking window mix selection (BLANKPULSEMIX). This is added for LLC topologies where blanking window settings need to be changed on the fly - providing greater configurability to do this.
- **Trip-Zone Submodule Enhancements:** A CAPEVENT signal can generate a CBC or One-shot trip event.
- **Diode Emulation Submodule:** The diode emulation mode was added to provide hardware features and the necessary hooks into other IPs to implement a robust diode mode sense and control in a noisy environment.
- **Minimum Dead-Band and Illegal Combo Logic Submodule:** The minimum dead-band logic was added to provide the ability to configure the minimum dead-band duration between a complimentary set of ePWMs.

  To detect and make sure that under no circumstances, the ePWM states result in potentially hazardous combinations, a Look Up Table (LUT) has been added that can be used to re-configure the ePWM outputs.
- **Event Trigger Submodule Enhancements:** To enable unevenly spaced over-sampling of the ePWM period, the event trigger module trigger select is modified such that multiple events can trigger SOCA, SOCB, and INT events (ETINTMIX).

The ePWM Type 4 is functionally compatible to Type 2 (a Type 3 does not exist). Type 4 has the following enhancements in addition to the Type 2 features:

- **Register Address Map:** Additional registers are required for new features on ePWM Type 4. The ePWM register address space has been remapped for better alignment and easy usage.
- **Delayed Trip Functionality:** Changes have been added to achieve deadband insertion capabilities to support, for example, delayed trip functionality needed for peak current mode control type application scenarios. This has been accomplished by allowing comparator events to go into the Action Qualifier as a trigger event (Events T1 and T2). If comparator T1 / T2 events are used to edit the PWM, changes to the PWM waveform do not take place immediately. Instead, the waveform synchronizes to the next TBCLK.
- **Dead-Band Generator Submodule Enhancements:** Shadowing of the DBCTL register to allow dynamic configuration changes.

- **One Shot and Global Load of Registers:** The ePWM Type 4 allows one shot and global load capability from shadow to active registers to avoid partial loads in, for example, multiphase applications. ePWM Type 4 also allows a programmable prescale of shadow to active load events. ePWM Type 4 Global Load can simplify ePWM software by removing interrupts and making sure that all registers are loaded at the same time.
- **Trip-Zone Submodule Enhancements:** Independent flags have been added to reflect the trip status for each of the TZ sources. Changes have been made to the trip-zone submodule to support certain power converter switching techniques like valley switching.
- **Digital Compare Submodule Enhancements:** Blanking window filter register width has been increased from 8 to 16 bits. DCCAP functionality has been enhanced to provide more programmability.
- **PWM SYNC Related Enhancements:** The ePWM Type 4 allows PWM SYNCOUT generation based on CMPC and CMPD events. These events can also be used for PWMSYNC pulse selection.

The ePWM Type 2 is fully compatible to Type 1. Type 2 has the following enhancements in addition to the Type 1 features:

- **High-Resolution Dead-Band Capability:** High-resolution capability is added to dead-band RED and FED in half-cycle clocking mode.
- **Dead-Band Generator Submodule Enhancements:** The ePWM Type 2 has features to enable both RED and FED on either PWM outputs. Provides increased dead band with 14-bit counters and dead-band / dead-band high-resolution registers are shadowed
- **High-Resolution Extension available on ePWMxB outputs:** Provides the ability to enable high-resolution period and duty cycle control on ePWMxB outputs. This is discussed in more detail in Section 30.18.
- **Counter Compare Submodule Enhancements:** The ePWM Type 2 allows interrupts and SOC events to be generated by additional counter compares CMPC and CMPD.
- **Event Trigger Submodule Enhancements:** Prescaling logic to issue interrupt requests and ADC start of conversion expanded up to every 15 events. This submodule allows software initialization of event counters on SYNC event.
- **Digital Compare Submodule Enhancements:** Digital Compare Trip Select logic [DCTRIPSEL] has up to 12 external trip sources selected by the Input X-BAR logic in addition to an ability to OR all of them (up to 14 [external and internal sources]) to create the respective DCxEVTs.
- **Simultaneous Writes to TBPRD and CMPx Registers:** This feature allows writes to TBPRD, CMPA:CMPAHR, CMPB:CMPBHR, CMPC and CMPD of any ePWM module to be tied to any other ePWM module, and also allows all ePWM modules to be tied to a particular ePWM module if desired.
- **Shadow to Active Load on SYNC of TBPRD and CMP Registers:** This feature supports simultaneous writes of TBPRD and CMPA/B/C/D registers.

An effective PWM peripheral must be able to generate complex pulse width waveforms with minimal CPU overhead or intervention and must be highly programmable and very flexible while being easy to understand and use. The ePWM unit described here addresses these requirements by allocating all needed timing and control resources on a per PWM channel basis. Cross coupling or sharing of resources has been avoided; instead, the ePWM is built up from smaller single channel submodules with separate resources that can operate together as required to form a system. This modular approach results in an orthogonal architecture and provides a more transparent view of the peripheral structure, helping users to understand the operation quickly.

In this document, the letter x within a signal or submodule name is used to indicate a generic ePWM instance on a device. For example, output signals EPWMxA and EPWMxB refer to the output signals from the ePWMx instance. Thus, EPWM1A and EPWM1B belong to ePWM1 and likewise EPWM4A and EPWM4B belong to ePWM4.

Type0 to Type1 Enhancements

- **Increased Dead-Band Resolution:** Dead-band clocking has been enhanced to allow half-cycle clocking to double resolution.
- **Enhanced Interrupt and SOC Generation:** Interrupts and ADC start-of-conversion can now be generated on both the TBCTR == zero and TBCTR == period events. This feature enables dual edge PWM control.

Additionally, the ADC start-of-conversion can be generated from an event defined in the digital compare submodule.

- **High-Resolution Period Capability:** Provides the ability to enable high-resolution period. This is discussed in more detail in Section 30.18.
- **Digital Compare Submodule:** The digital compare submodule enhances the event triggering and trip zone submodules by providing filtering, blanking and improved trip functionality to digital compare signals. Such features are essential for peak current mode control and for support of analog comparators.

Note

The name of the sync signal that goes to the CMPSS has been updated from PWMSYNC to EPWMSYNCPER (SYNCPER/PWMSYNCPER/EPWMxSYNCPER) to avoid confusion with the other EPWM sync signals EPWMSYNCI and EPWMSYNCO. For a description of these signals, see Table 30-2.

30.1.1 EPWM Related Collateral

Foundational Materials

- C28x Academy - EPWM
- C29x Academy - Enhanced Pulse Width Modulation (EPWM)
- *Real-Time Control Reference Guide*
  - Refer to the EPWM section

Getting Started Materials

- *C2000 ePWM Developer's Guide Application Report*
- Enhanced Pulse Width Modulator (ePWM) Training for C2000 MCUs (Video)
- *Flexible PWMs Enable Multi-Axis Drives, Multi-Level Inverters Application Report*
- Getting Started with the C2000 ePWM Module (Video)
- *Using PWM Output as a Digital-to-Analog Converter on a TMS320F280x Digital Signal Control Application Report*
  - Chapters 1 to 6 are Fundamental material, derivations, and explanations that are useful for learning about how PWM can be used to implement a DAC. Subsequent chapters are Getting Started and Expert material for implementing in a system.
- *Using the Enhanced Pulse Width Modulator (ePWM) Module Application Report*

Expert Materials

- C2000 real-time microcontrollers - Reference designs
  - See TI designs related to specific end applications used.
- *Leverage New Type ePWM Features for Multiple Phase Control Application Report*

30.1.2 Submodule Overview

The ePWM module represents one complete PWM channel composed of two PWM outputs: EPWMxA and EPWMxB. Multiple ePWM modules are instanced within a device as shown in Figure 30-1. Each ePWM instance is identical with one exception. Some instances include a hardware extension that allows more precise control of the PWM outputs. This extension is the high-resolution pulse width modulator (HRPWM) and is described in Section 30.18. See the device data sheet to determine which ePWM instances include this feature. Each ePWM module is indicated by a numerical value starting with 1. For example, ePWM1 is the first instance and ePWM3 is the third instance in the system and ePWMx indicates any instance.

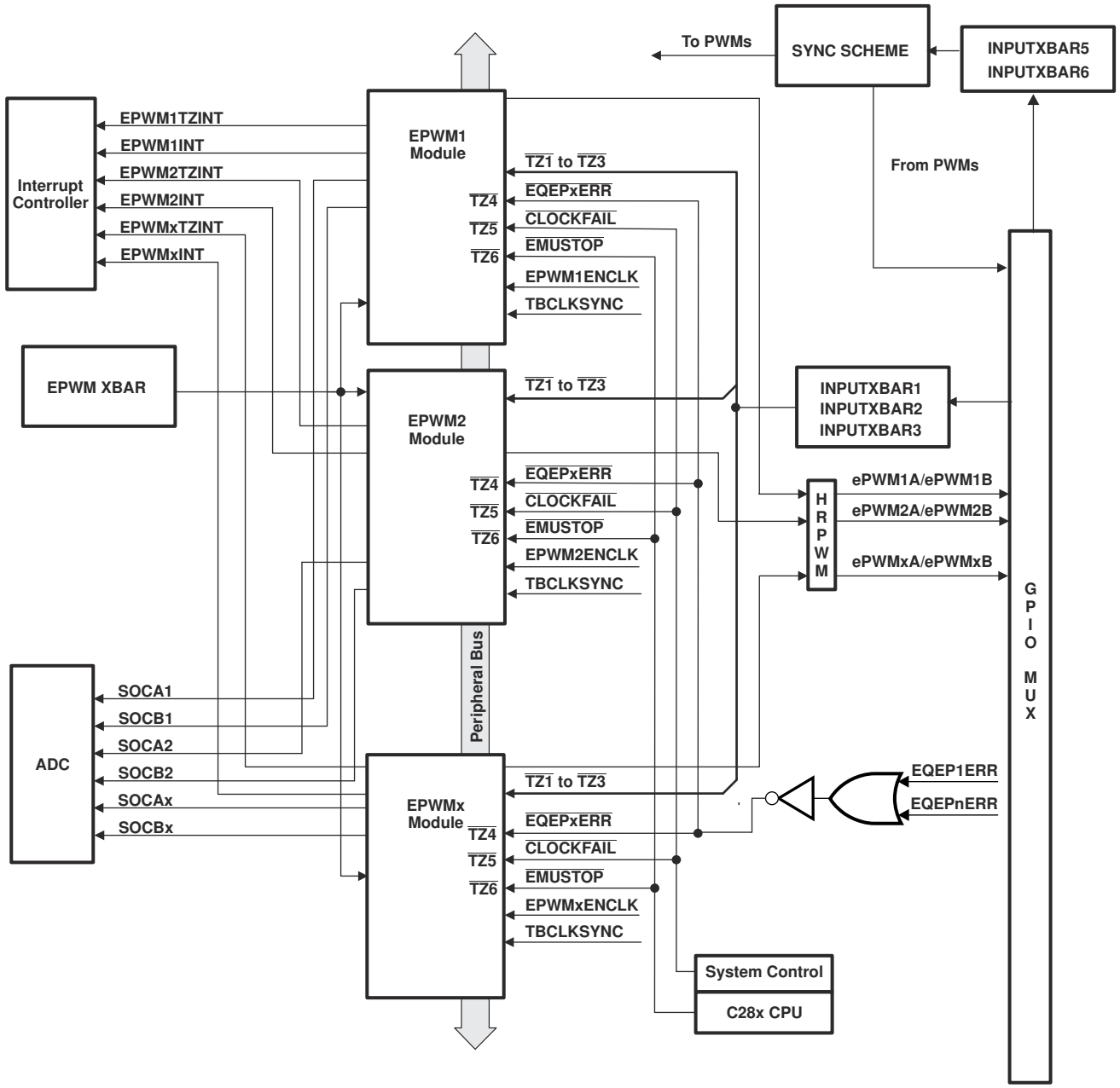
The ePWM modules are chained together by way of a clock synchronization scheme that allows them to operate as a single system when required. Additionally, this synchronization scheme can be extended to the capture peripheral submodules (eCAP). The number of submodules is device-dependent and based on target application needs. Submodules can also operate standalone.

Each ePWM module supports the following features:

- Dedicated 16-bit time-base counter with period and frequency control
- Two PWM outputs (EPWMxA and EPWMxB) that can be used in the following configurations:
  - Two independent PWM outputs with single-edge operation
  - Two independent PWM outputs with dual-edge symmetric operation
  - One independent PWM output with dual-edge asymmetric operation
- Asynchronous override control of PWM signals through software.
- Programmable phase-control support for lag or lead operation relative to other ePWM modules.
- Hardware-locked (synchronized) phase relationship on a cycle-by-cycle basis.
- Dead-band generation with independent rising and falling edge delay control.
- Programmable trip zone allocation of both cycle-by-cycle trip and one-shot trip on fault conditions.
- A trip condition can force either high, low, or high-impedance state logic levels at PWM outputs.
- All events can trigger both CPU interrupts and ADC start of conversion (SOC)
- Programmable event prescaling minimizes CPU overhead on interrupts.
- PWM chopping by high-frequency carrier signal, useful for pulse transformer gate drives.

Each ePWM module is connected to the input/output signals shown in Figure 30-1. The signals are described in detail in subsequent sections.

The order in which the ePWM modules are connected can differ from what is shown in Figure 30-1. See Section 30.4.3.3 for the synchronization scheme for a particular device. Each ePWM module consists of eight submodules and is connected within a system by way of the signals shown in Figure 30-2.



A. This signal exists only on devices with an eQEP submodule.

Figure 30-1. Multiple ePWM Modules

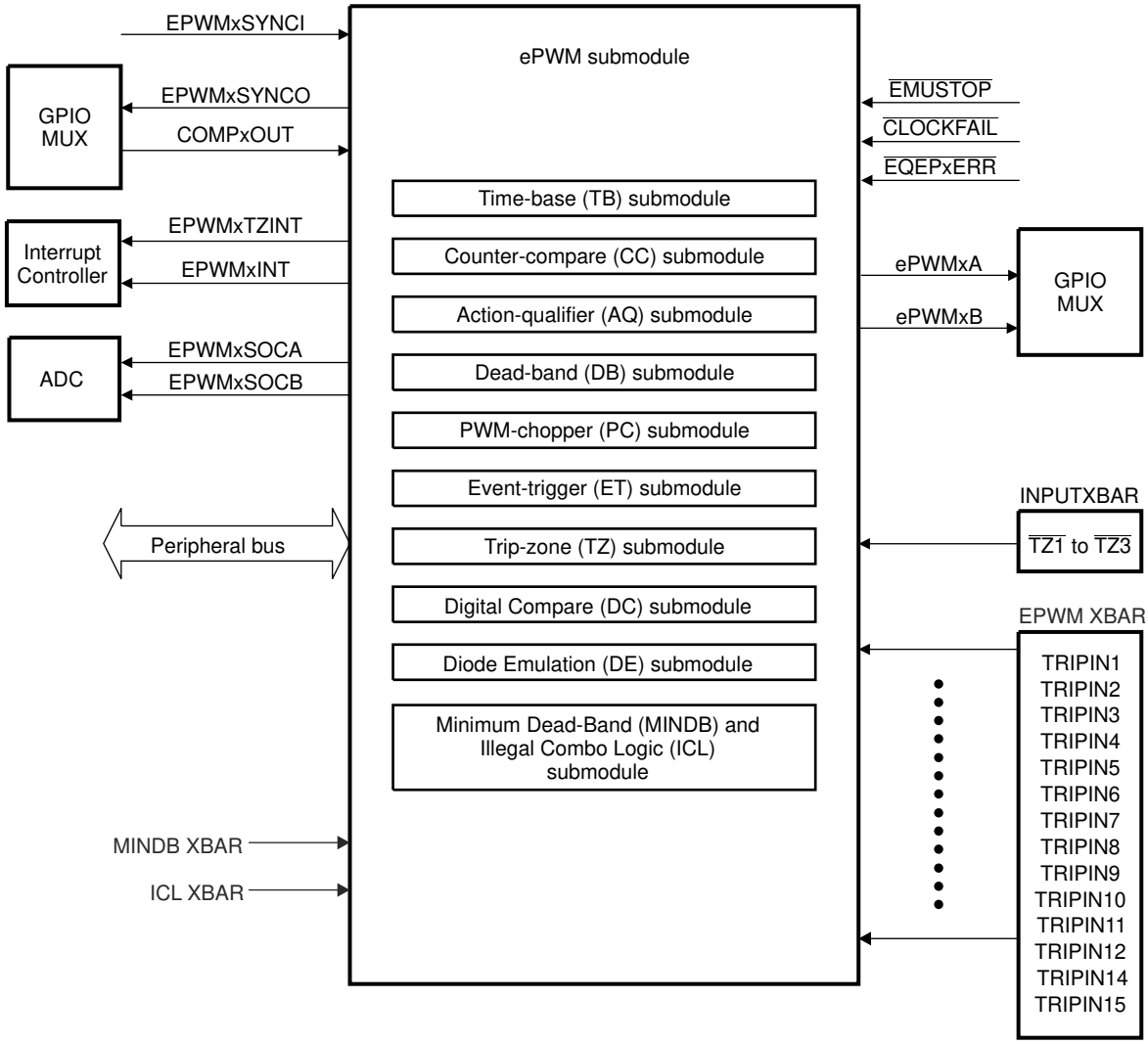
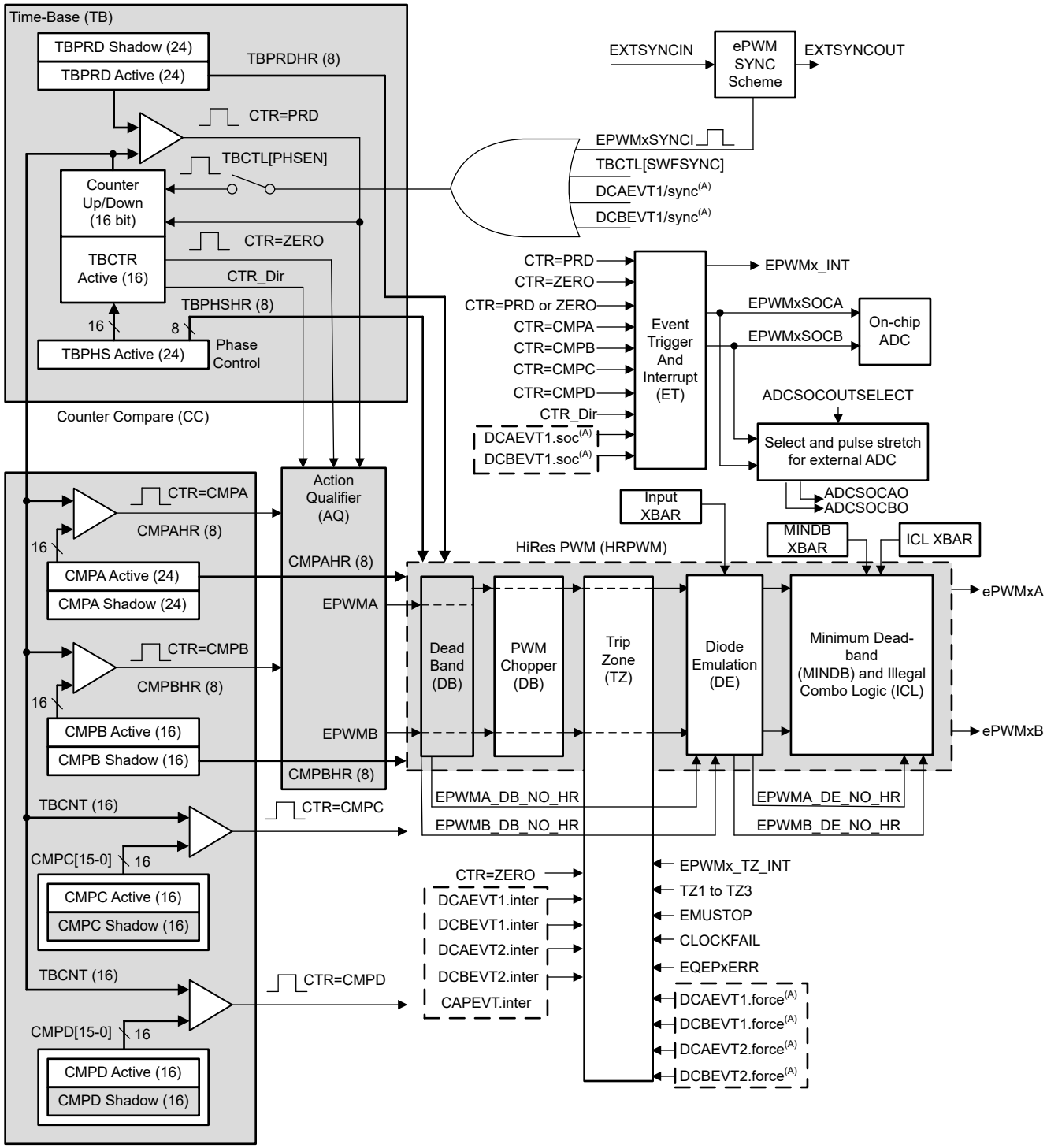


Figure 30-2. Submodules and Signal Connections for an ePWM Module

Figure 30-3 shows more internal details of a single ePWM module. The main signals used by the ePWM module are:

- **PWM output signals (EPWMxA and EPWMxB):** The PWM output signals are made available external to the device.
- **Trip-zone signals ($\overline{\text{TZ1}}$ to $\overline{\text{TZ6}}$):** These input signals alert the ePWM module of fault conditions external to the ePWM module. Each submodule on a device can be configured to either use or ignore any of the trip-zone signals. The $\overline{\text{TZ1}}$ to $\overline{\text{TZ3}}$ trip-zone signals can be configured as asynchronous inputs through the GPIO peripheral using the Input X-BAR logic, refer to Figure 30-75. $\overline{\text{TZ4}}$ is connected to an inverted EQEPx error signal (EQEPxERR), which can be generated from any one of the EQEP submodule (for those devices with an EQEP module). $\overline{\text{TZ5}}$ is connected to the system clock fail logic, and $\overline{\text{TZ6}}$ is connected to the $\overline{\text{EMUSTOP}}$ output from the CPU. This allows configuring a trip action when the clock fails or the CPU halts.
- **Time-base synchronization input (EPWMxSYNCI), output (EPWMxSYNCO), and peripheral (EPWMxSYNCPER) signals:** Each ePWM module can be synchronized with other ePWM modules or other peripherals, using EPWMSYNCINSEL. Each ePWM module can also generate a synchronization output signal. The source of the EPWMxSYNCOUT can be selected and enabled by EPWMSYNCOUTEN and TBCTL2.OSHTSYNCMODE. For more information, see Section 30.4.3.3.

  Each ePWM module also generates another PWMSYNC signal called EPWMxSYNCPER. EPWMxSYNCPER goes to the GPDAC and CMPSS for synchronization purposes. Functionality is configured using the HRPCTL register, but has no relation with the HRPWM. For more information on how EPWMxSYNCPER is used by the GPDAC and CMPSS, see the respective chapters.
- **ADC start-of-conversion signals (EPWMxSOCA and EPWMxSOCB):** Each ePWM module has two ADC start of conversion signals. Any ePWM module can trigger a start of conversion. Whichever event triggers the start of conversion is configured in the event-trigger submodule of the ePWM.
- **Comparator output signals (COMPxOUT):** Output signals from the comparator module can be fed through the Input X-BAR and EPWM X-BAR to one or all of the 15 trip inputs [TRIPIN1-TRIPIN15] and in conjunction with the trip zone signals can generate digital compare events.
- **Peripheral bus:** The peripheral bus is 32-bits wide and allows both 16-bit and 32-bit writes to the ePWM register file.



A. These events are generated by the ePWM Digital Compare (DC) submodule based on the levels of the TRIPIN inputs.

Figure 30-3. ePWM Modules and Critical Internal Signal Interconnects

30.2 Configuring Device Pins

To connect the device input pins to the module, the Input X-BAR and EPWM X-BAR must be used. Some examples of when an external signal can be needed are TZx, TRIPx, and EXTSYNCIN. Any GPIO on the device can be configured as an input. The GPIO input qualification can be set to asynchronous mode by setting the appropriate GPxQSEL register bits to 11b. The internal pullups can be configured in the GPyPUD register. Since the GPIO mode is used, the GPyINV register can invert the signals.

The GPIO mux registers must be configured for this peripheral. To avoid glitches on the pins, the GPyGMUX bits must be configured first (while keeping the corresponding GPyMUX bits at the default of zero), followed by writing the GPyMUX register to the desired value.

See the *General-Purpose Input/Output (GPIO)* chapter for more details on GPIO mux, GPIO settings, and XBAR configuration.

30.3 ePWM Modules Overview

Ten submodules are included in every ePWM peripheral. Each of these submodules performs specific tasks that can be configured by software.

Table 30-1 lists the key submodules together with a list of the main configuration parameters. For example, if you need to adjust or control the duty cycle of a PWM waveform, see the counter-compare submodule in Section 30.5 for relevant details.

Table 30-1. Submodule Configuration Parameters

Submodule	Configuration Parameter or Option
Time Base (TB)	• Scale the time-base clock (TBCLK) relative to the ePWM clock (EPWMCLK). • Configure the PWM time-base counter (TBCTR) frequency or period. • Set the mode for the time-base counter: – count-up mode: used for asymmetric PWM – count-down mode: used for asymmetric PWM – count-up-and-down mode: used for symmetric PWM • Configure the time-base phase relative to another ePWM module. • Synchronize the time-base counter between modules through hardware or software. • Configure the direction (up or down) of the time-base counter after a synchronization event. • Simultaneous writes to the TBPRD registers on all PWM's corresponding to the configuration on EPWMXLINK. • Configure how the time-base counter behaves when the device is halted by an emulator. • Specify the source for the synchronization output of the ePWM module • Configure one shot and global load of registers in this module.
Counter Compare (CC)	• Specify the PWM duty cycle for output EPWMxA and output EPWMxB • Specify the time at which switching events occur on the EPWMxA or EPWMxB output • Specify the programmable delay for interrupt and SOC generation with additional comparators • Simultaneous writes to the CMPA, CMPB, CMPC, CMPD registers on all PWM's corresponding to the configuration on EPWMXLINK. • Configure one shot and global load of registers in this module. • Generate up to four pulses in one ePWM period through the complex waveform (XCMP) mode feature

Table 30-1. Submodule Configuration Parameters (continued)

Submodule	Configuration Parameter or Option
Action Qualifier (AQ)	• Specify the type of action taken when a time-base counter-compare, trip-zone submodule, or comparator event occurs: – No action taken – Output EPWMxA and EPWMxB switched high – Output EPWMxA and EPWMxB switched low – Output EPWMxA and EPWMxB toggled • Force the PWM output state through software control • Configure and control the PWM dead band through software • Configure one shot and global load of registers in this module.
Dead-Band Generator (DB)	• Control of traditional complementary dead-band relationship between upper and lower switches • Specify the output rising-edge-delay value • Specify the output falling-edge delay value • Bypass the dead-band module entirely. In this case the PWM waveform is passed through without modification. • Option to enable half-cycle clocking for double resolution. • Allow ePWMxB phase shifting with respect to the ePWMxA output. • Configure one shot and global load of registers in this module. • Simultaneous writes to the DBRED, DBREDHR, DBFED, DBFEDHR registers on all PWM's corresponding to the configuration on EPWMXLINK2.
PWM Chopper (PC)	• Create a chopping (carrier) frequency. • Pulse width of the first pulse in the chopped pulse train. • Duty cycle of the second and subsequent pulses. • Bypass the PWM chopper module entirely. In this case the PWM waveform is passed through without modification.
Trip Zone (TZ)	• Configure the ePWM module to react to one, all, or none of the trip-zone signals or digital compare events. • Specify the trip action taken when a fault occurs: – Force EPWMxA and EPWMxB high – Force EPWMxA and EPWMxB low – Force EPWMxA and EPWMxB to a high-impedance state – Configure EPWMxA and EPWMxB to ignore any trip condition. • Configure how often the ePWM reacts to each trip-zone signal: – One-shot – Cycle-by-cycle • Enable the trip-zone to initiate an interrupt. • Bypass the trip-zone module entirely. • Programmable option for cycle-by-cycle trip clear • If desired, independently configure trip actions taken when time-base counter is counting down.
Diode Emulation(DE)	• Choose any of the comparator outputs as trips to detect entry into DE mode. • Monitor the DE mode duration and generate a trip event to PWMs. • Ability to switch the comparator thresholds, dynamically in hardware upon DE mode entry. • Cycle-by-cycle and one-shot modes of clearing/de-evaluating the DE condition.
Minimum Dead-Band (MINDB) and Illegal Combo Logic (ICL)	• Add a minimum amount of delay between ePWM channels • Define non-supported output combinations and drive output high or low if combination occurs
Event Trigger (ET)	• Enable the ePWM events that trigger an interrupt. • Enable ePWM events that trigger an ADC start-of-conversion event. • Specify the rate at which events cause triggers (every occurrence or every 2nd or up to 15th occurrence) • Poll, set, or clear event flags

Table 30-1. Submodule Configuration Parameters (continued)

Submodule	Configuration Parameter or Option
Digital Compare (DC)	• Enables comparator (COMP) module outputs and trip zone signals which are configured using the Input X-BAR to create events and filtered events • Specify event-filtering options to capture TBCTR counter, generate blanking window, or insert delay in PWM output or time-base counter based on captured value.

30.4 Time-Base (TB) Submodule

Each ePWM module has a time-base submodule that determines all of the event timing for the ePWM module. Built-in synchronization logic allows the time-base of multiple ePWM modules to work together as a single system.

Figure 30-4 illustrates the time-base submodule within the ePWM.

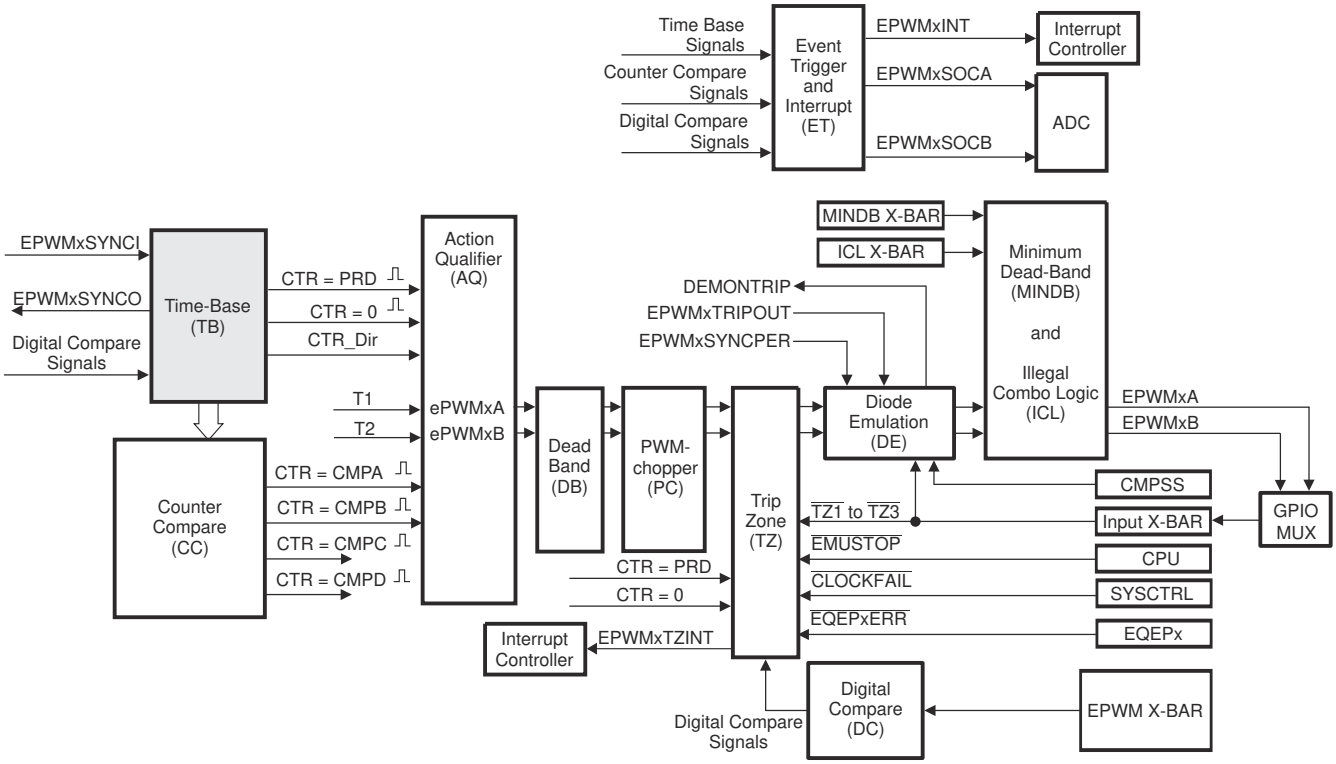


Figure 30-4. Time-Base Submodule

30.4.1 Purpose of the Time-Base Submodule

The time-base submodule can be configured for the following:

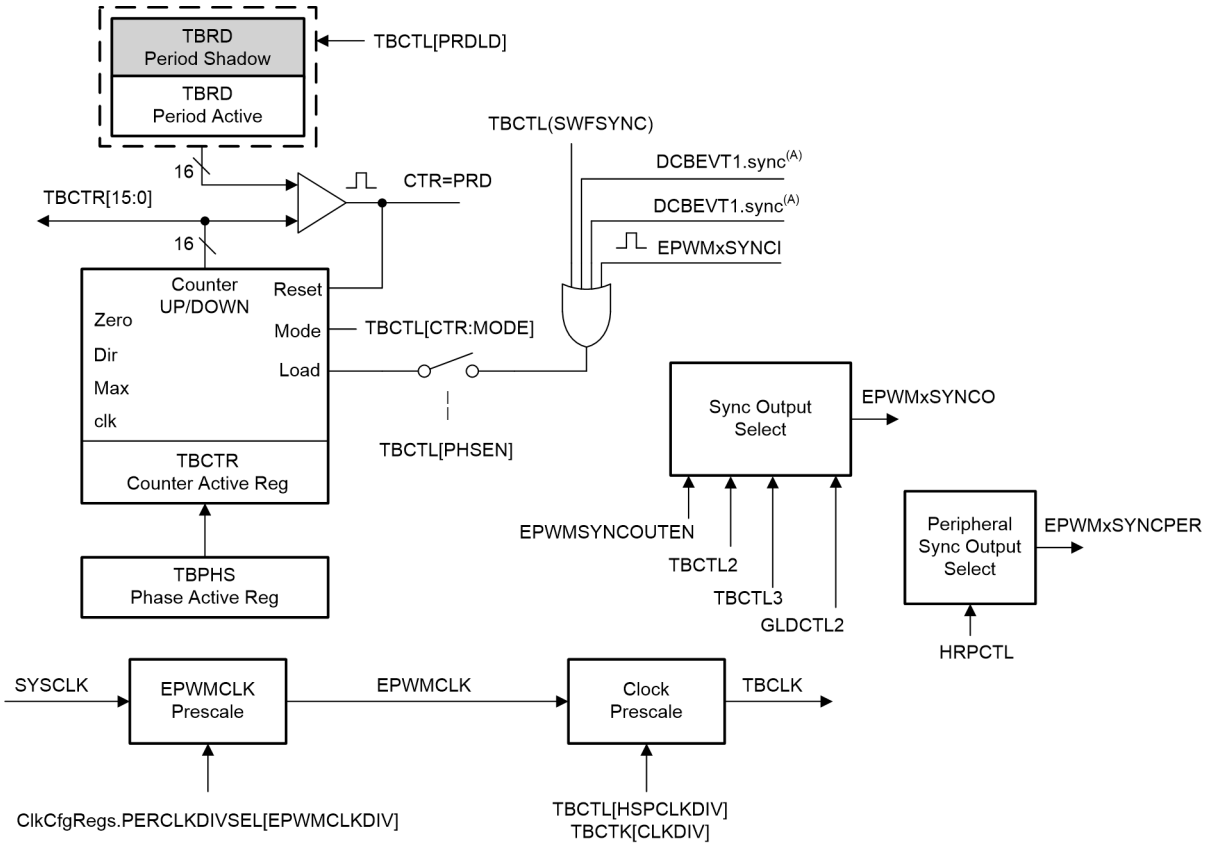
- Specify the ePWM time-base counter (TBCTR) frequency or period to control how often events occur.
- Manage time-base synchronization with other ePWM modules.
- Maintain a phase relationship with other ePWM modules.
- Set the time-base counter to count-up, count-down, or count-up-and-down mode.
- Generate the following events:
  - CTR = PRD: Time-base counter equal to the specified period (TBCTR = TBPRD).
  - CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00).
- Configure the rate of the time-base clock; a prescaled version of the ePWM clock (EPWMCLK). This allows the time-base counter to increment/decrement at a slower rate.

Note

If required by the application code to update the TBCTR value through software while the TBCTR is counting, note that the time-base module needs at least 1 TBCLK cycle for the time-base related events to be realized. Hence, the TBCTR can be written with TBCTR = PRD-1 instead of TBCTR = PRD (in case the counter is counting up) and can be written as TBCTR = 1 instead of TBCTR = 0 (in case the counter is counting down) for the events to be realized.

30.4.2 Controlling and Monitoring the Time-Base Submodule

The block diagram in Figure 30-5 shows the critical signals and registers of the time-base submodule. Table 30-2 provides descriptions of the key signals associated with the time-base submodule.



A. These signals are generated by the digital compare (DC) submodule.

Figure 30-5. Time-Base Submodule Signals and Registers

Table 30-2. Key Time-Base Signals

Signal	Description
EPWMxSYNCI	Time-base synchronization input. Input pulse used to synchronize the time-base counter with the counter of other ePWM modules. For more information on all of the signals available for synchronization, see EPWMSYNCINSEL. For information on the synchronization order of a particular device, see Section 30.4.3.3.
EPWMxSYNCO	Time-base synchronization output. This output pulse is used to synchronize the counter of other ePWM modules. Using EPWMSYNCOUTEN, TBCTL2, TBCTL3 and GLDCTL2, the source of the output pulse is selected.
EPWMxSYNCPER	Time-base peripheral synchronization output. This output signal is used to synchronize the GPDAC and CMPSS to the EPWM. The output signal can be configured using the HRPCTL register. Note that this signal has no relation with the HRPWM.
CTR = PRD	Time-base counter equal to the specified period. This signal is generated whenever the counter value is equal to the active period register value. That is when TBCTR = TBPRD.
CTR = Zero	Time-base counter equal to zero This signal is generated whenever the counter value is zero. That is when TBCTR equals 0x00.
CTR = CMPB	Time-base counter equal to active counter-compare B register (TBCTR = CMPB). This event is generated by the counter-compare submodule and used by the synchronization out logic
CTR_dir	Time-base counter direction. Indicates the current direction of the ePWM's time-base counter. The signal is high when the counter is increasing and the signal is low when the counter is decreasing.
CTR_max	Time-base counter equal max value. (TBCTR = 0xFFFF) Generated event when the TBCTR value reaches the maximum value. This signal is only used only as a status bit
TBCLK	Time-base clock. This is a prescaled version of the ePWM clock (EPWMCLK) and is used by all submodules within the ePWM. This clock determines the rate at which time-base counter increments or decrements.

30.4.3 Calculating PWM Period and Frequency

The frequency of PWM events is controlled by the time-base period (TBPRD) register and the mode of the time-base counter. Figure 30-6 shows the period (T_{pwm}) and frequency (F_{pwm}) relationships for the up-count, down-count, and up-down-count time-base counter modes when the period is set to 4 (TBPRD = 4). The time increment for each step is defined by the time-base clock (TBCLK) which is a prescaled version of the ePWM clock (EPWMCLK).

The time-base counter has three modes of operation selected by the time-base control register (TBCTL):

- **Up-Down Count Mode:** In up-down count mode, the time-base counter starts from zero and increments until the period (TBPRD) value is reached. When the period value is reached, the time-base counter then decrements until the counter reaches zero. At this point, the counter repeats the pattern and begins to increment.
- **Up-Count Mode:** In up-count mode, the time-base counter starts from zero and increments until the counter reaches the value in the period register (TBPRD). When the period value is reached, the time-base counter resets to zero and begins to increment once again.
- **Down-Count Mode:** In down-count mode, the time-base counter starts from the period (TBPRD) value and decrements until the counter reaches zero. When the counter reaches zero, the time-base counter is reset to the period value and begins to decrement once again.

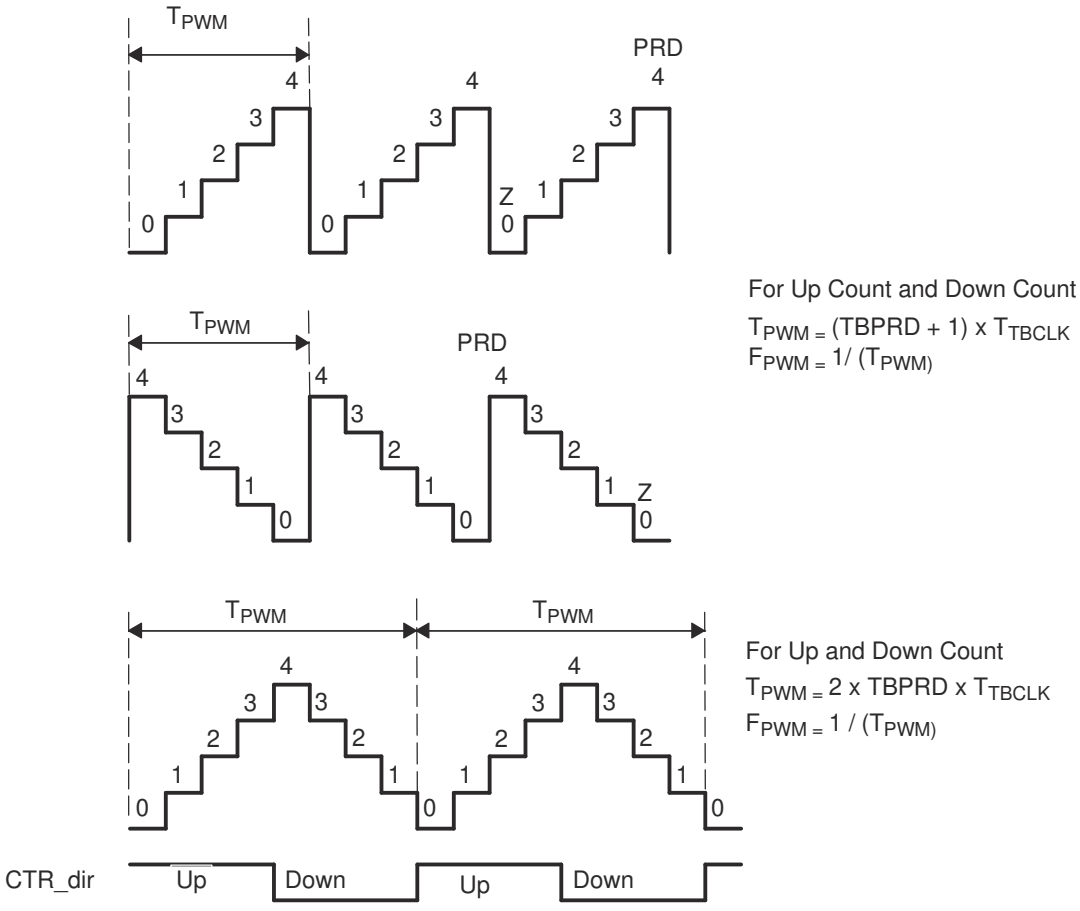


Figure 30-6. Time-Base Frequency and Period

30.4.3.1 Time-Base Period Shadow Register

The time-base period register (TBPRD) has a shadow register. Shadowing allows the register update to be synchronized with the hardware. The following definitions are used to describe all shadow registers in the ePWM module:

- **Active Register:** The active register controls the hardware and is responsible for actions that the hardware causes or invokes.
- **Shadow Register:** The shadow register buffers provide a temporary holding location for the active register and have no direct effect on any control hardware. At a strategic point in time, the shadow register content is transferred to the active register. This prevents corruption or spurious operation due to the register being asynchronously modified by software.

The memory address of the shadow period register is the same as the active register. Which register is written to or read from is determined by the TBCTL[PRDLD] bit. This bit enables and disables the TBPRD shadow register as follows:

- **Time-Base Period Shadow Mode:** The TBPRD shadow register is enabled when TBCTL[PRDLD] = 0. Reads from and writes to the TBPRD memory address go to the shadow register. The shadow register contents are transferred to the active register (TBPRD (Active) ← TBPRD (shadow)) when the time-base counter equals zero (TBCTR = 0x00) and/or a sync event as determined by the TBCTL2[PRDLDSYNC] bit. The PRDLDSYNC bit is valid only if TBCTL[PRDLD] = 0. By default the TBPRD shadow register is enabled. The sources for the SYNC input is explained in Section 30.4.3.3.

  The global load control mechanism can also be used with the time-base period register by configuring the appropriate bits in the global load configuration register (GLDCFG). When global load mode is selected the transfer of contents from shadow register to active register, for all registers that have this mode enabled, occurs at the same event as defined by the configuration bits in Global Shadow to Active Load Control Register (GLDCTL). Global load control mechanism is explained in Section 30.4.7.
- **Time-Base Period Immediate Load Mode:** If immediate load mode is selected (TBCTL[PRDLD] = 1), then a read from or a write to the TBPRD memory address goes directly to the active register.

30.4.3.2 Time-Base Clock Synchronization

The TBCLKSYNC bit in the peripheral clock enable registers allows all users to globally synchronize all enabled ePWM modules to the time-base clock (TBCLK). When set, all enabled ePWM module clocks are started with the first rising edge of TBCLK aligned. For synchronized TBCLKs, the prescalers for each ePWM module must be set identically.

The proper procedure for enabling ePWM clocks is as follows:

1. Enable ePWM module clocks in the PCLKCRx register
2. Set TBCLKSYNC = 0
3. Configure ePWM modules
4. Set TBCLKSYNC = 1

In a multicore environment, GTBCLKSYNC can be used to override the core-specific TBCLKSYNC. When GTBCLKSYNC is set, TBCLKSYNC is ignored in all cores and therefore clearing or setting the TBCLKSYNC has no affect. If this feature is not required, GTBCLKSYNC must be cleared. In a multicore environment where different ePWM modules are assigned to different cores, the GTBCLKSYNC bit can be used to enable and disable the Time-Base clock of all ePWM modules simultaneously.

30.4.3.3 Time-Base Counter Synchronization

The ePWM synchronization scheme allows for increased flexibility of synchronization of the ePWM modules. Each ePWM module has a synchronization input (SYNCI), a synchronization output (SYNCO) and a peripheral synchronization output (SYNCPER). In Figure 30-7, EXTSYNCIN1 is sourced from INPUTXBAR5 and EXTSYNCIN2 is sourced from INPUTXBAR6, which can be configured to select any GPIO as the synchronization input.

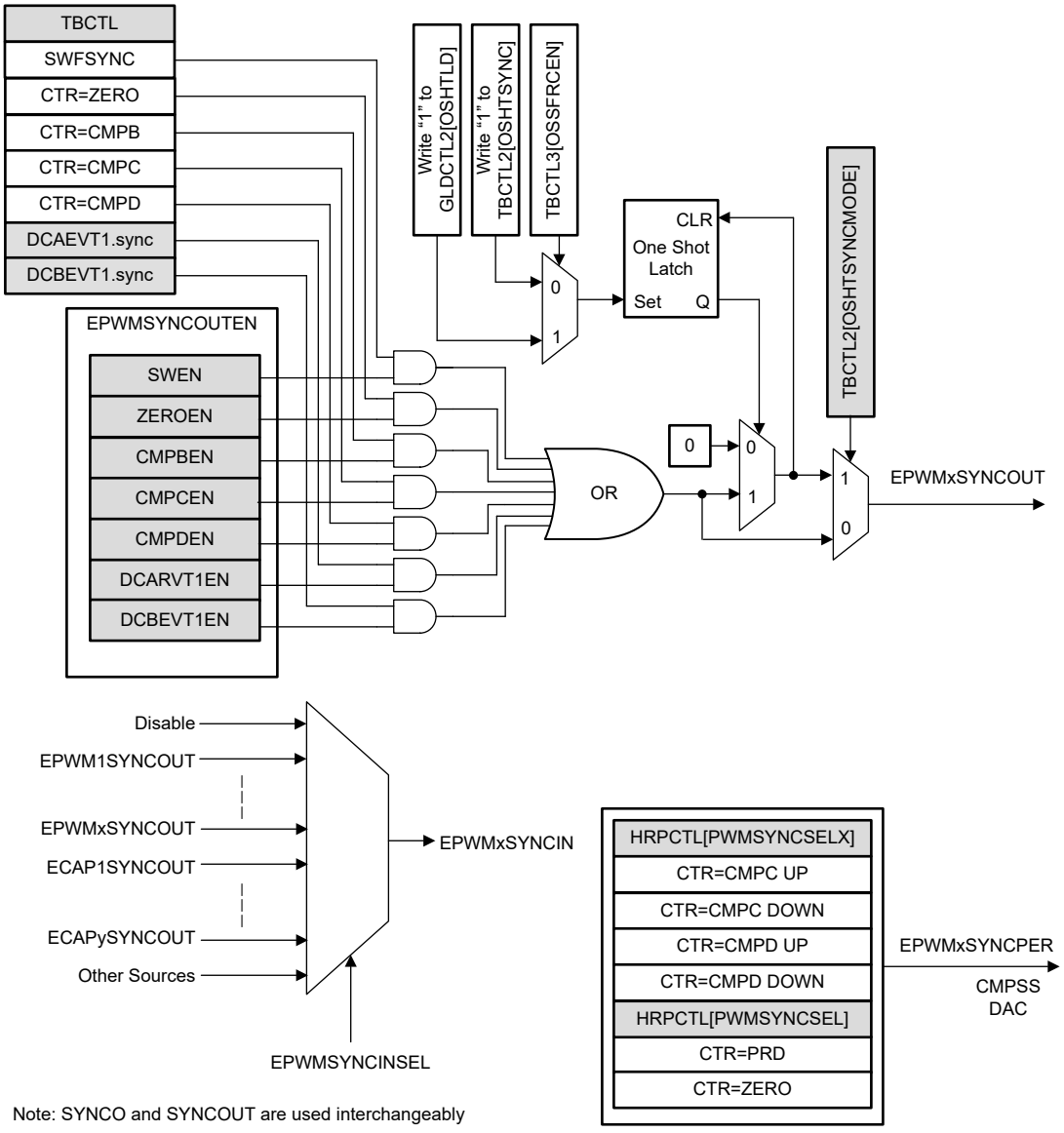


Figure 30-7. Time-Base Counter Synchronization Scheme

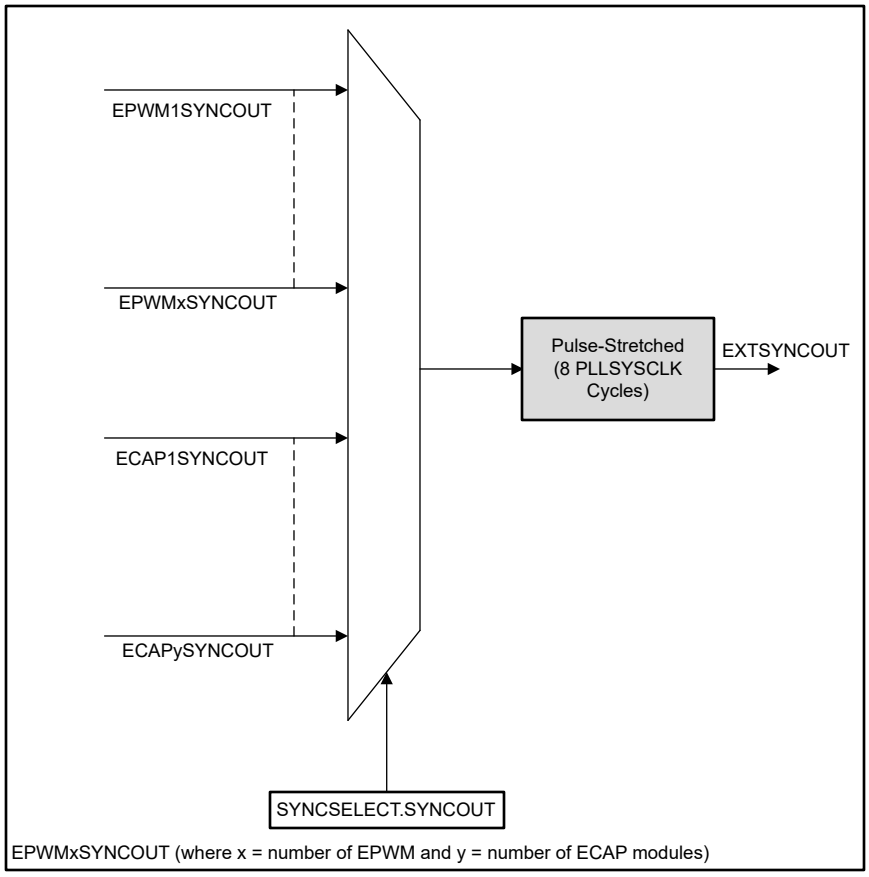


Figure 30-8. ePWM External SYNC Output

Note

See the data sheet for the number of ePWM and eCAP modules available on your specific device.

Each ePWM module can be configured to use or ignore the synchronization input. If the TBCTL[PHSEN] bit is set, then the time-base counter (TBCTR) of the ePWM module is automatically loaded with the phase register (TBPHS) contents when one of the following conditions occur:

- **EPWMxSYNCI: Synchronization Input Pulse:** The value of the phase register is loaded into the counter register when an input synchronization pulse is detected (TBPHS → TBCTR). This operation occurs on the next valid time-base clock (TBCLK) edge.

  The delay from internal control module to target modules is given by:
  - if (TBCLK = EPWMCLK): 2 x EPWMCLK
  - if (TBCLK < EPWMCLK): 1 x TBCLK
- **Software Forced Synchronization Pulse:** Writing a 1 to the TBCTL[SWFSYNC] control bit invokes a software forced synchronization. This pulse is ORed with the synchronization input signal, and therefore has the same effect as a pulse on EPWMxSYNCI.
- **Digital Compare Event Synchronization Pulse:** DCAEVT1 and DCBEVT1 digital compare events can be configured to generate synchronization pulses which have the same affect as EPWMxSYNCI.

Note

If the EPWMxSYNCI signal is held high, the sync does not continuously occur. The EPWMxSYNCI is rising edge activated. Don't use multiple edges in a PWM cycle if sync functionality is used.

Note

When modifying the TBPHS register during run-time, missed action qualifier events can occur due to sudden jumps in the TBCTR value at the time of the SYNCIN pulse. To recreate the behavior of missed action qualifier events, configure an action qualifier event on a T1 or T2 event on a SYNCIN event. The T1 or T2 action qualifier event must be enabled and disabled during runtime depending on the value of TBPHS.

This feature enables the ePWM module to be automatically synchronized to the time base of another ePWM module. Lead or lag phase control can be added to the waveforms generated by different ePWM modules to synchronize them. In up-down-count mode, the TBCTL[PHSDIR] bit configures the direction of the time-base counter immediately after a synchronization event. The new direction is independent of the direction prior to the synchronization event. The PHSDIR bit is ignored in count-up or count-down modes. See Figure 30-9 through Figure 30-12 for examples.

Clearing the TBCTL[PHSEN] bit configures the ePWM to ignore the synchronization input pulse.

30.4.3.4 ePWM SYNC Selection

Table 30-3 specifies the sources for the ePWM SYNC input and output.

Table 30-3. ePWM SYNC Selection

Index	Signal
0	EPWM_SYNC_DISABLE
1	EPWM1_SYNCOUT
2	EPWM2_SYNCOUT
3	EPWM3_SYNCOUT
4	EPWM4_SYNCOUT
5	EPWM5_SYNCOUT
6	EPWM6_SYNCOUT
7	EPWM7_SYNCOUT
8	EPWM8_SYNCOUT
9	EPWM9_SYNCOUT
10	EPWM10_SYNCOUT
11	EPWM11_SYNCOUT
12	EPWM12_SYNCOUT
13	EPWM13_SYNCOUT
14	EPWM14_SYNCOUT
15	EPWM15_SYNCOUT
16	EPWM16_SYNCOUT
17	ECAP1_SYNCOUT
18	ECAP2_SYNCOUT
19	ECAP3_SYNCOUT
20	ECAP4_SYNCOUT
21	ECAP5_SYNCOUT
22	ECAP6_SYNCOUT
23	ECAP7_SYNCOUT
24	INPUTXBAR5
25	INPUTXBAR6
26	ECAT_SYNC0
27	ECAT_SYNC1
28	EPWM17_SYNCOUT

Table 30-3. ePWM SYNC Selection (continued)

Index	Signal
29	EPWM18_SYNCOUT
30	FSIRXA_TRIG1
31	FSIRXB_TRIG1
32	FSIRXC_TRIG1
33	FSIRXD_TRIG1
34-63	Reserved

30.4.4 Phase Locking the Time-Base Clocks of Multiple ePWM Modules

The TBCLKSYNC bit can be used to globally synchronize the time-base clocks of all enabled ePWM modules on a device. This bit is part of the device's clock enable registers and is described in the *System Control and Interrupts* section of this manual. When TBCLKSYNC = 0, the time-base clock of all ePWM modules is stopped (default). When TBCLKSYNC = 1, all ePWM time-base clocks are started with the rising edge of TBCLK aligned. For synchronized TBCLKs, the prescaler bits in the TBCTL register of each ePWM module must be set identically. The proper procedure for enabling the ePWM clocks is:

1. Enable the individual ePWM module clocks. This is described in the *System Control and Interrupts* chapter.
2. Set TBCLKSYNC = 0. This stops the time-base clock within any enabled ePWM module.
3. Configure the prescaler values and desired ePWM modes.
4. Set TBCLKSYNC = 1.

30.4.5 Simultaneous Writes Between ePWM Register Instances

For variable frequency applications, there is a need for simultaneous writes of TBPRD and CMPx registers between ePWM modules. This prevents situations where a CTR = 0 or CTR = PRD pulse forces a shadow to active load of these registers before all registers are updated between ePWM modules (resulting in some registers being loaded from new shadow values while others are loaded from old shadow values). To support this, an ePWM instance linking scheme for all registers in an ePWM instance has been added. There is now a dedicated XLINK region in the memory map to support writes to any of the ePWM registers to reflect in other ePWM instances. All ePWM individual module instances are mirrored to the XLINK region by configuring the SYSCTL'S EPWMXLINKCFG bit. If multiple ePWM instances have this feature enabled, then any writes in the XLINK region of an ePWM instance reflects to all other ePWM instances that have the ePWM instances enabled. If the write occurs in the individual register memory space, then the update only reflects on that individual ePWM. Any reads to mirrored regions return 0.

Note

Writes to any location that has no ePWM instance mapped to the location still mirror in all locations that are enabled.

Example:

Linking PWMs 1,2,3 using Driverlib will result in

SysCtl_enableEPWMXLINK(EPWM1BASE | EPWM2BASE | EPWM3BASE);

Write to the XLINK region of PWM 1 to write to EPWM 1,2,3 using the XLINK_BASE address results in

EPWM_setCounterCompareValue(EPWM1XLINK_BASE, EPWM_COUNTER_COMPARE_A, CMPAVAL);

Any writes to the EPWMnXLINK_BASE memory region will write to all other PWMs. If you only want to update an independent PWM's register value then do not use the XLINK base memory mapped regions. Below is an example to write to EPWM1's CMPA value independently of other linked PWM 2,3.

EPWM_setCounterCompareValue(EPWM1_BASE, EPWM_COUNTER_COMPARE_A, CMPAVAL);

30.4.6 Time-Base Counter Modes and Timing Waveforms

The time-base counter operates in one of four modes:

- Up-count mode that is asymmetrical
- Down-count mode that is asymmetrical
- Up-down-count that is symmetrical
- Frozen where the time-base counter is held constant at the current value

To illustrate the operation of the first three modes, the following timing diagrams show when events are generated and how the time-base responds to an EPWMxSYNCI signal.

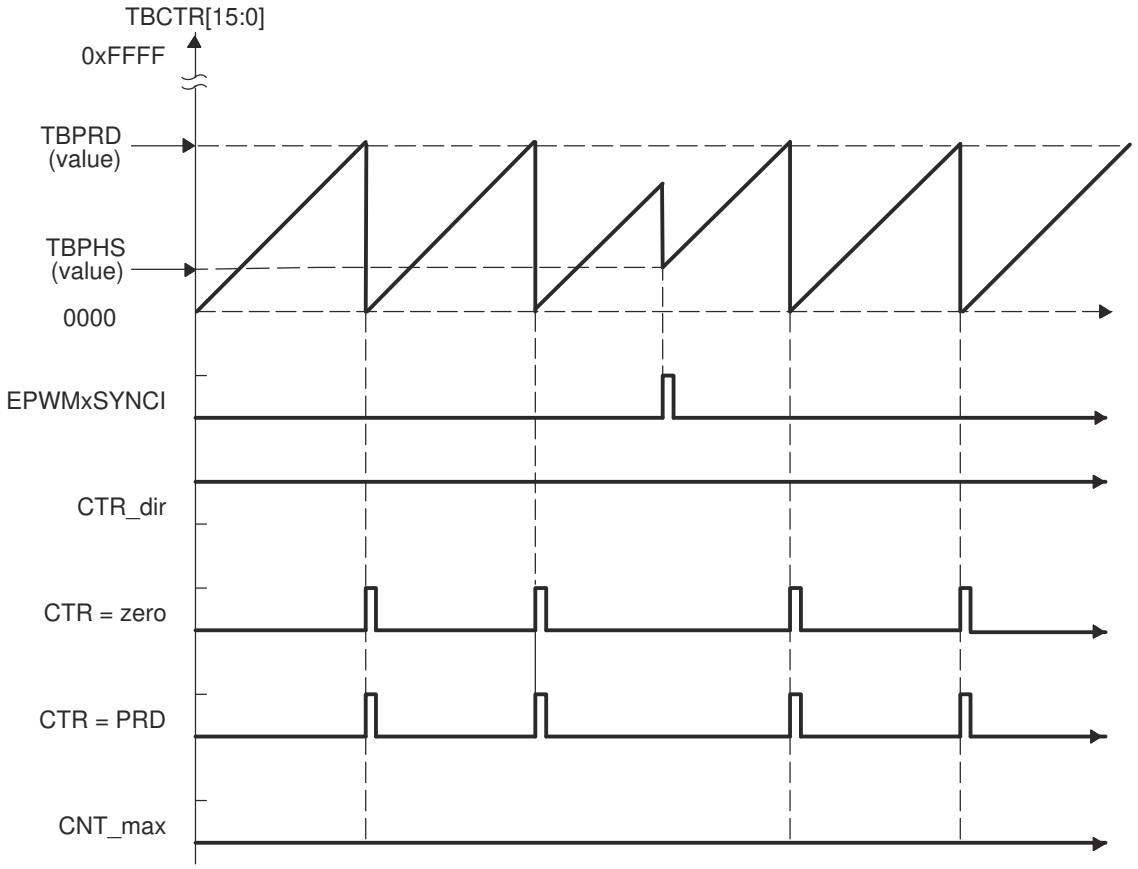


Figure 30-9. Time-Base Up-Count Mode Waveforms

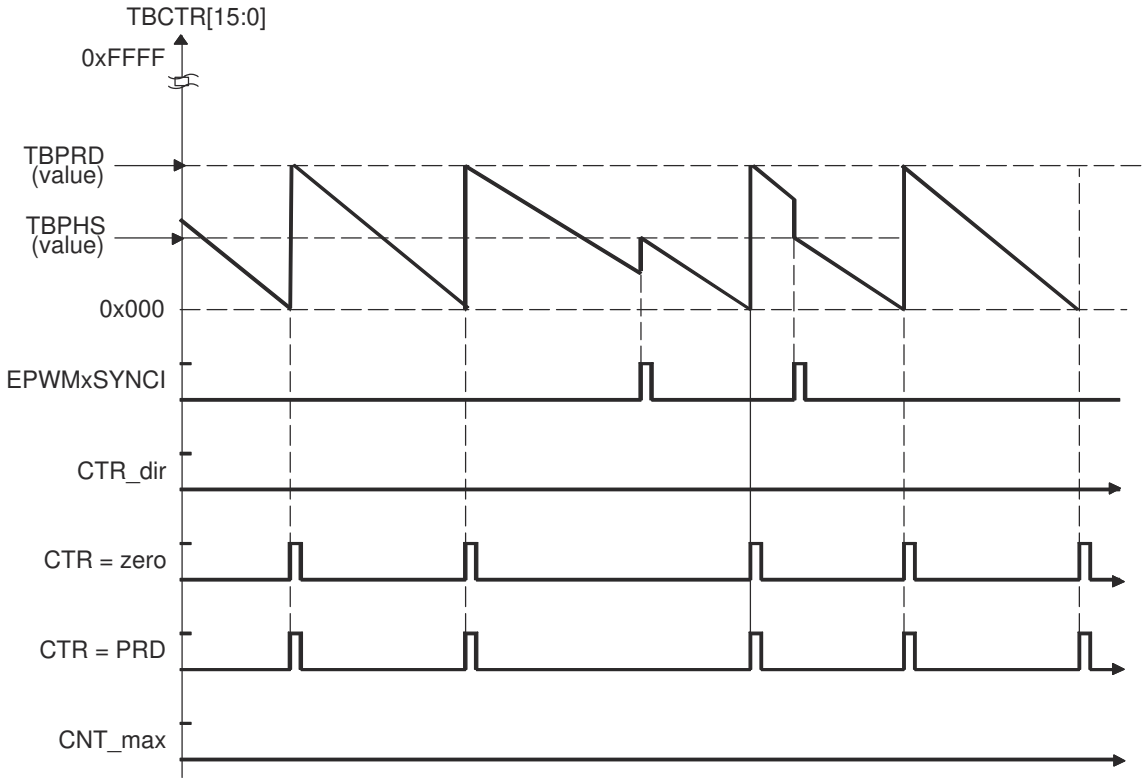


Figure 30-10. Time-Base Down-Count Mode Waveforms

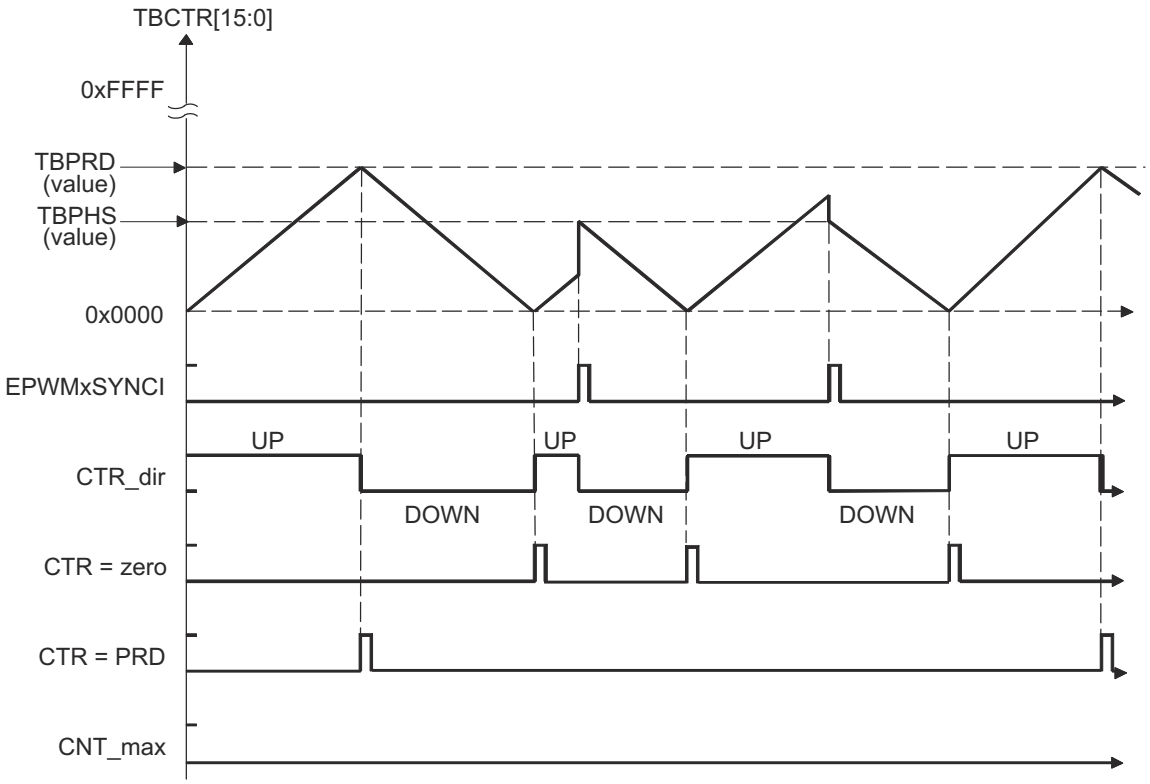


Figure 30-11. Time-Base Up-Down-Count Waveforms, TBCTL[PHSDIR = 0] Count Down On Synchronization Event

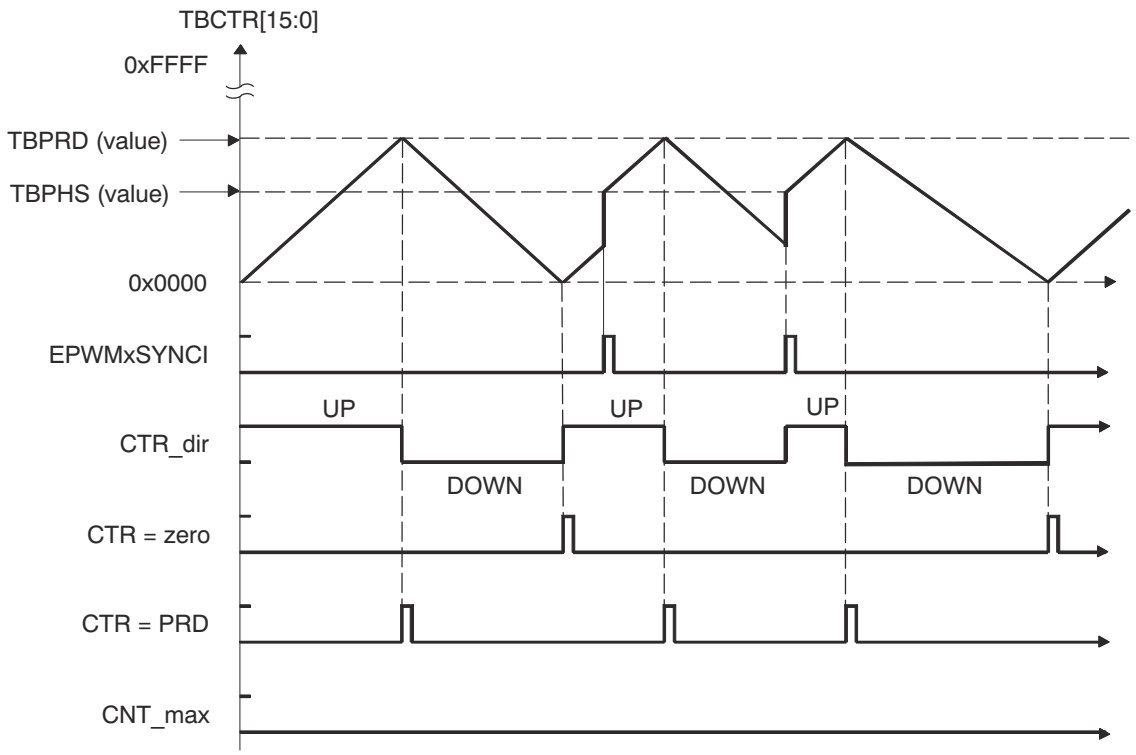


Figure 30-12. Time-Base Up-Down Count Waveforms, TBCTL[PHSDIR = 1] Count Up On Synchronization Event

30.4.7 Global Load

Figure 30-13 shows the signals and registers associated with the global load feature.

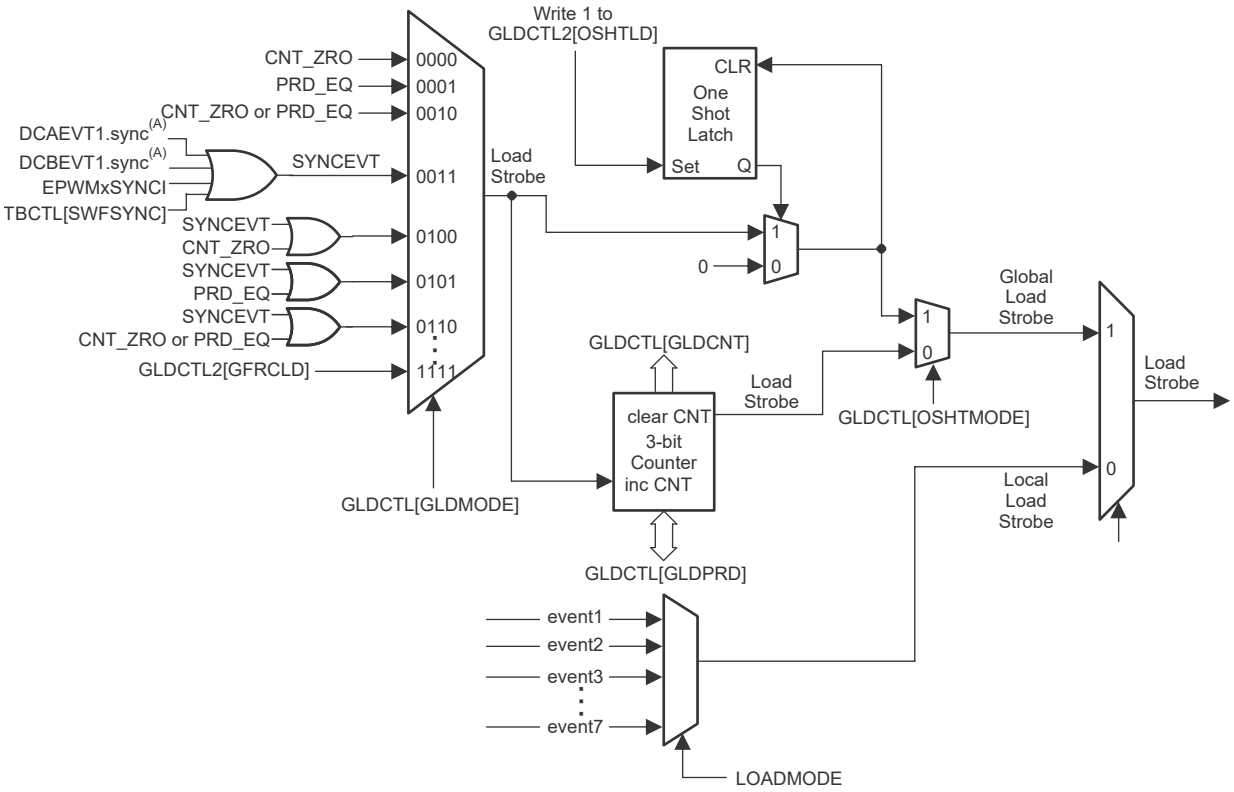


Figure 30-13. Global Load: Signals and Registers

Note

The SYNCEVT signal is only propagated through when PHSEN is SET.

When this feature is enabled, the transfer of contents from the shadow register to the active register, for all registers that have this mode enabled, occurs at the same event as defined by the configuration bits in Global Shadow to Active Load Control Register (GLDCTL[GLDMODE]). When GLDCTL[GLD] = 1, shadow to active load event selection bits for individual shadowed registers are ignored and global load mode takes effect for the corresponding registers enabled by GLDCFG[REGx].

When GLDCTL[GLD] = 1 and GLDCFG[REGx] = 0, global load mode does not affect the corresponding register (REGx). Shadow to active load event selection bits for individual shadowed registers decide how the transfer of contents from shadow register to active register takes place.

30.4.7.1 Global Load Pulse Pre-Scalar

This feature provides the capability to choose shadow to active transfers to happen once in 'N' occurrences of selected global load pulse (GLDCTL[GLDMODE]). This pre-scale functionality is not available for registers that cannot or are not configured to use the global load mechanism (that is, GLDCTL[GLD] = 0 or GLDCFG[REGx] = 0).

30.4.7.2 One-Shot Load Mode

This feature allows users to cause the shadow register to active register transfers to occur once. When GLDCTL2[OSHTLD] = 1 the shadow to active register transfer, for registers that are configured to use the global load mechanism, takes place on the event selected by GLDCTL[GLDMODE].

Software force loading of contents from shadow register to active register is possible by using GLDCTL2[GFRCLD]. The GLDCTL2 register can also be linked across multiple PWM modules by using EPWMXLINK[GLDCTL2LINK]. This, along with the one-shot load mode feature discussed above, provides a method to correctly update multiple PWM registers in one or more PWM modules at certain PWM events or, if desired, in the same clock cycle. This is very useful in variable frequency applications and/or multi-phase interleaved applications.

30.4.7.3 One-Shot Sync Mode

To enable the one-shot sync mode to generate a SYNCOUT pulse, configure the TBCTL2[OSHTSYNCMODE] bit and set the TBCTL2[OSHTSYNC] bit as shown in Figure 30-14.

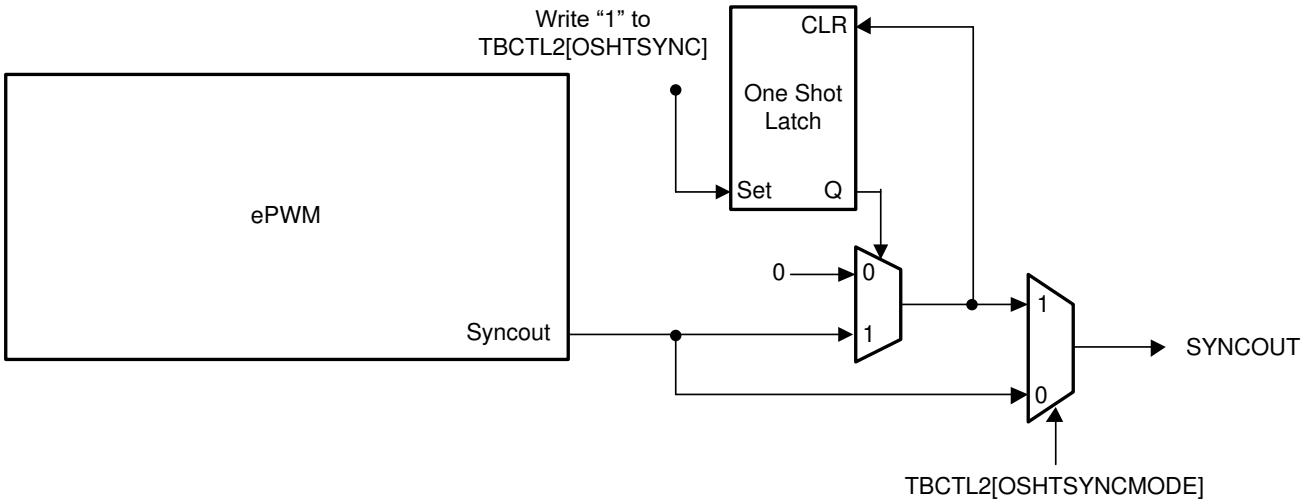


Figure 30-14. One-Shot Sync Mode

30.5 Counter-Compare (CC) Submodule

Figure 30-15 illustrates the counter-compare submodule within the ePWM.

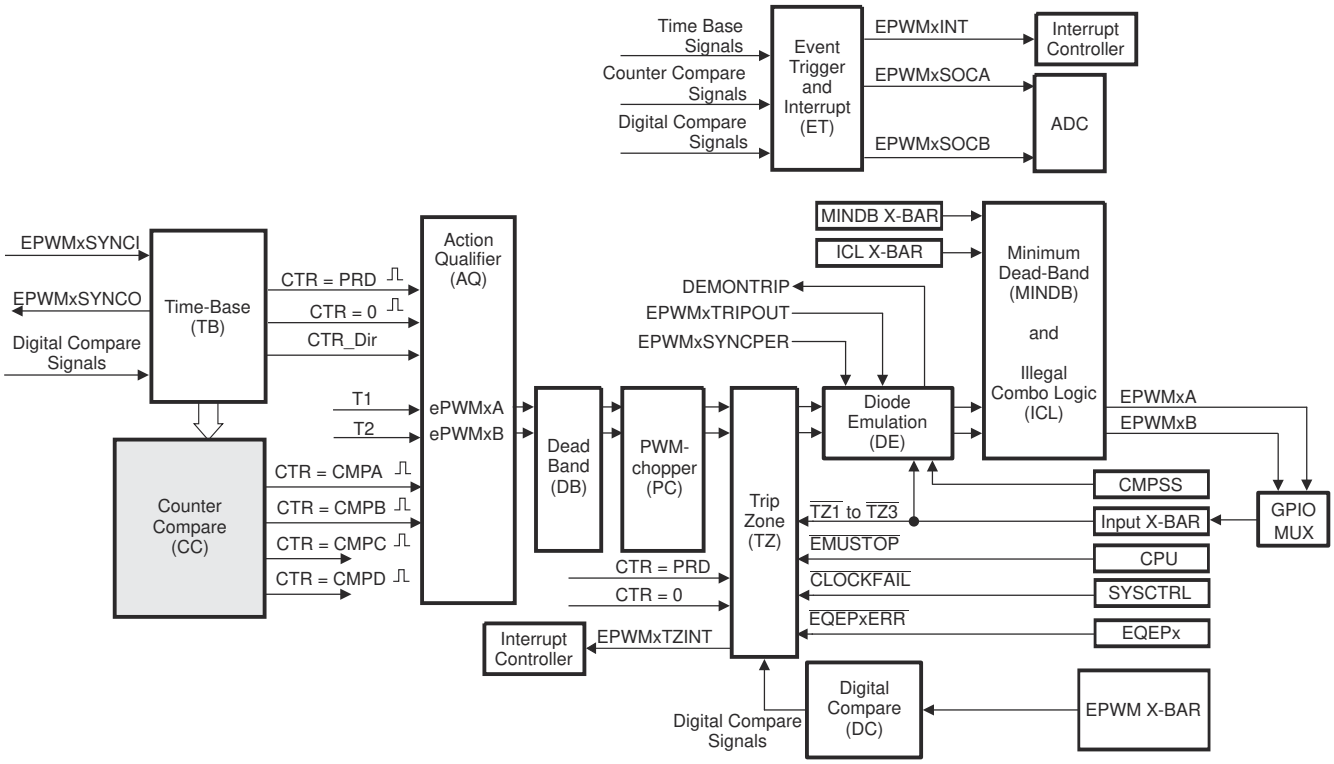


Figure 30-15. Counter-Compare Submodule

30.5.1 Purpose of the Counter-Compare Submodule

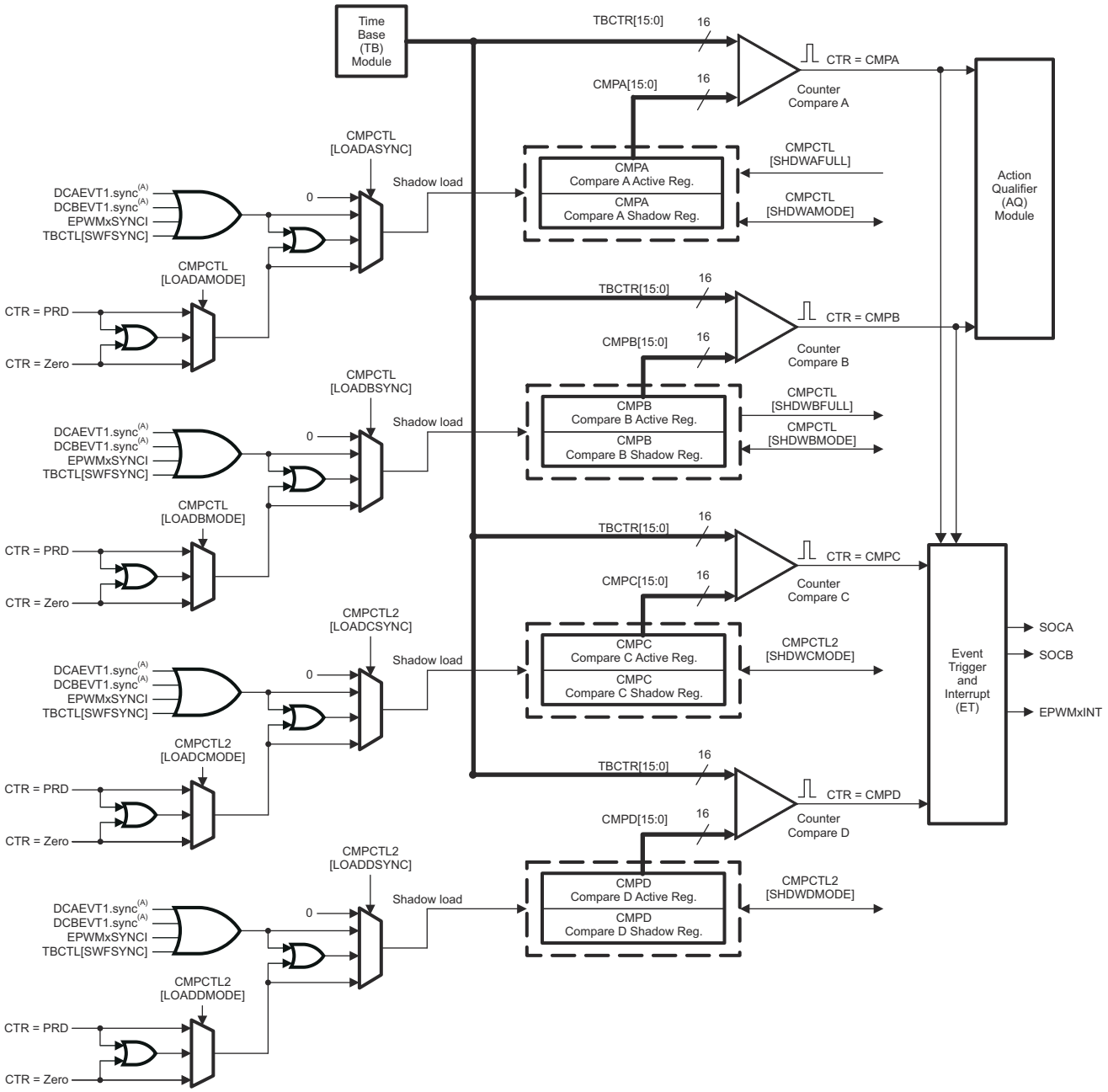
The counter-compare submodule takes as input the time-base counter value. This value is continuously compared to the counter-compare A (CMPA), counter-compare B (CMPB), counter-compare C (CMPC), and counter-compare D (CMPD) registers. When the time-base counter is equal to one of the compare registers, the counter-compare unit generates an appropriate event.

The counter-compare:

- Generates events based on programmable time stamps using the CMPA, CMPB, CMPC, and CMPD registers:
  - CTR = CMPA: Time-base counter equals counter-compare A register (TBCTR = CMPA)
  - CTR = CMPB: Time-base counter equals counter-compare B register (TBCTR = CMPB)
  - CTR = CMPC: Time-base counter equals counter-compare C register (TBCTR = CMPC)
  - CTR = CMPD: Time-base counter equals counter-compare D register (TBCTR = CMPD)
- Controls the PWM duty cycle, if the action-qualifier submodule is configured appropriately using counter-compare A (CMPA) and counter-compare B (CMPB)
- Shadows new compare values to prevent corruption or glitches during the active PWM cycle

30.5.2 Controlling and Monitoring the Counter-Compare Submodule

The counter-compare submodule operation is shown in Figure 30-16.



A. These events are generated by the ePWM digital compare (DC) submodule based on the levels of the TRIPIN inputs (for example, CMPSSx and $\overline{TZ}$ signals).

Figure 30-16. Detailed View of the Counter-Compare Submodule

30.5.3 Operational Highlights for the Counter-Compare Submodule

The counter-compare submodule is responsible for generating events that can be used in the action-qualifier and event-trigger submodules. There are four independent compare events:

1. CTR = CMPA: Time-base counter equal to counter-compare A register (TBCTR = CMPA).
2. CTR = CMPB: Time-base counter equal to counter-compare B register (TBCTR = CMPB).
3. CTR = CMPC: Time-base counter equal to counter-compare C register (TBCTR = CMPC). This event can be used to generate an event in the event trigger submodule only.
4. CTR = CMPD: Time-base counter equal to counter-compare D register (TBCTR = CMPD). This event can be used to generate an event in the event trigger submodule only

For up-count or down-count mode, each event occurs only once per cycle. For up-down count mode, each event occurs twice per cycle if the compare value is between 0x00-TBPRD; and once per cycle if the compare value is equal to 0x00 or equal to TBPRD. These events are applies to the action-qualifier submodule where the events are qualified by the counter direction and converted into actions if enabled. Refer to Section 30.6.1 for more details.

The counter-compare registers CMPA and CMPB each have an associated shadow register. Shadowing provides a way to keep updates to the registers synchronized with the hardware. When shadowing is used, updates to the active registers only occur at strategic points. This prevents corruption or spurious operation due to the register being asynchronously modified by software. The memory address of the active register and the shadow register is identical. The register that is written to or read from is determined by the CMPCTL[SHDWAMODE] and CMPCTL[SHDWBMODE] bits. These bits enable and disable the CMPC shadow register and CMPD shadow register, respectively. The behavior of the two load modes is:

Shadow Mode:

The shadow mode for the CMPA is enabled by clearing the CMPCTL[SHDWAMODE] bit and the shadow register for CMPB is enabled by clearing the CMPCTL[SHDWBMODE] bit. Shadow mode is enabled by default for both CMPA and CMPB.

If the shadow register is enabled then the content of the shadow register is transferred to the active register on one of the following events as specified by the CMPCTL[LOADAMODE], CMPCTL[LOADBMODE], CMPCTL[LOADASYNC], and CMPCTL[LOADBSYNC] register bits:

- CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD).
- CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00)
- Both CTR = PRD and CTR = Zero
- SYNC event caused by DCAEVT1 or DCBEVT1 or EPWMxSYNCI or TBCTL[SWFSYNC]
- Both SYNC event or a selection made by LOADAMODE/LOADBMODE

Only the active register contents are used by the counter-compare submodule to generate events to be sent to the action-qualifier.

Note

Refer to Section 30.6.5 for valid configurations of CMPA/CMPB and LOADAMODE/LOADBMODE.

Immediate Load Mode:

If the immediate load mode is selected (that is, CMPCTL[SHDWAMODE] = 1 or CMPCTL[SHDWBMODE] = 1), then a read from or a write to the register goes directly to the active register.

Additional Comparators

The counter-compare submodule on ePWMs type 2 and later are responsible for generating two additional independent compare events based on two compare registers, which is fed to Event Trigger submodule:

1. CTR = CMPC: Time-base counter equal to counter-compare C register (TBCTR = CMPC).
2. CTR = CMPD: Time-base counter equal to counter-compare D register (TBCTR = CMPD).

The counter-compare registers CMPC and CMPD each have an associated shadow register. By default this register is shadowed. The memory address of the active register and the shadow register is identical. The value in the active CMPC and CMPD register is compared to the time-base counter (TBCTR). When the values are equal, the counter compare module generates a “time-base counter equal to counter compare C or counter compare D ” event respectively. Shadowing of this register is enabled and disabled by the CMPCTL2[SHDWCMODE] and CMPCTL2[SHDWDMODE] bit. These bits enable and disable the CMPC shadow register and CMPD shadow register respectively. The behavior of the two load modes is described below:

Shadow Mode:

The shadow mode for the CMPC is enabled by clearing the CMPCTL2[SHDWCMODE] bit and the shadow register for CMPD is enabled by clearing the CMPCTL2[SHDWDMODE] bit. Shadow mode is enabled by default for both CMPC and CMPD.

If the shadow register is enabled then the content of the shadow register is transferred to the active register on one of the following events as specified by the CMPCTL2[LOADCMODE], CMPCTL2[LOADDMODE], CMPCTL2[LOADCSYNC], and CMPCTL2[LOADDSYNC] register bits:

- CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD).
- CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00)
- Both CTR = PRD and CTR = Zero
- SYNC event caused by DCAEVT1 or DCBEVT1 or EPWMxSYNCI or TBCTL[SWFSYNC]
- Both SYNC event or a selection made by LOADCMODE/LOADDMODE

Only the active register contents are used by the counter-compare submodule to generate events to be sent to the action-qualifier.

Immediate Load Mode:

If the immediate load mode is selected (that is, CMPCTL2[SHDWCMODE] = 1 or CMPCTL2[SHDWDMODE] = 1), then a read from or a write to the register goes directly to the active register.

Global Load Support

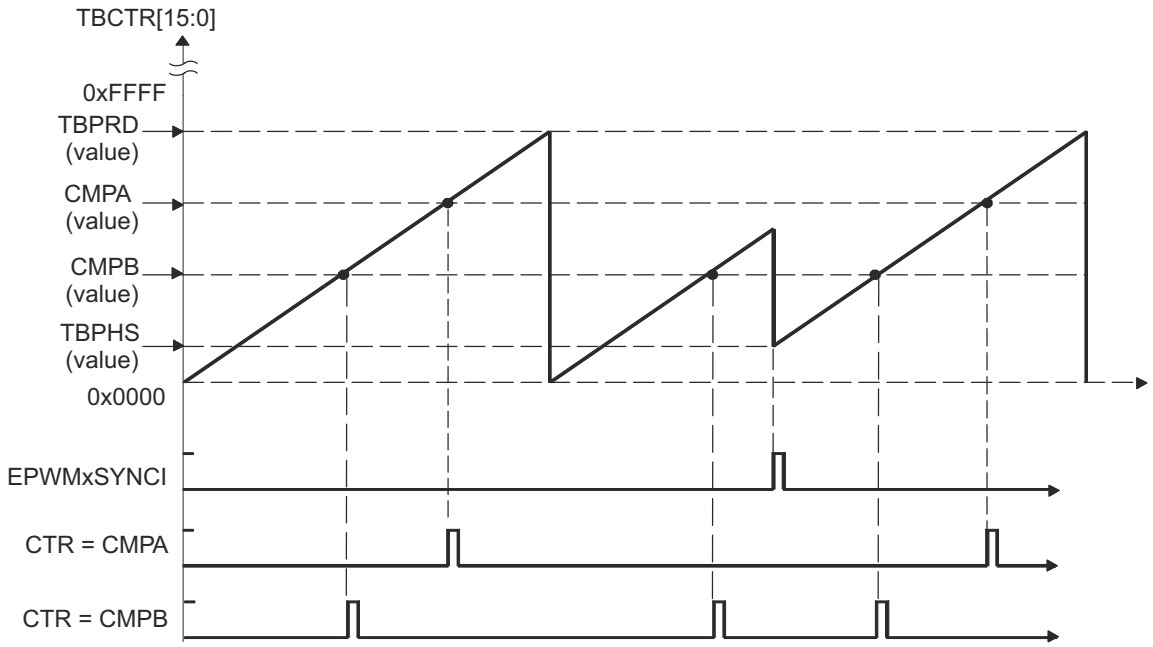
The global load control mechanism can also be used for all counter-compare registers by configuring the appropriate bits in the global load configuration register (GLDCFG). When the global load mode is selected the transfer of contents from shadow register to active register, for all registers that have this mode enabled, occurs at the same event as defined by the configuration bits in the Global Shadow to Active Load Control Register (GLDCTL). The global load control mechanism is explained in Section 30.4.7.

30.5.4 Count Mode Timing Waveforms

The counter-compare module can generate compare events in all three count modes:

- Up-count mode: used to generate an asymmetrical PWM waveform.
- Down-count mode: used to generate an asymmetrical PWM waveform.
- Up-down-count mode: used to generate a symmetrical PWM waveform.

To best illustrate the operation of the first three modes, the timing diagrams in Figure 30-17 through Figure 30-20 show when events are generated and how the EPWMxSYNCI signal interacts.



An EPWMxSYNCI external synchronization event can cause a discontinuity in the TBCTR count sequence. This can lead to a compare event being skipped. This skipping is considered normal operation and must be taken into account.

Figure 30-17. Counter-Compare Event Waveforms in Up-Count Mode

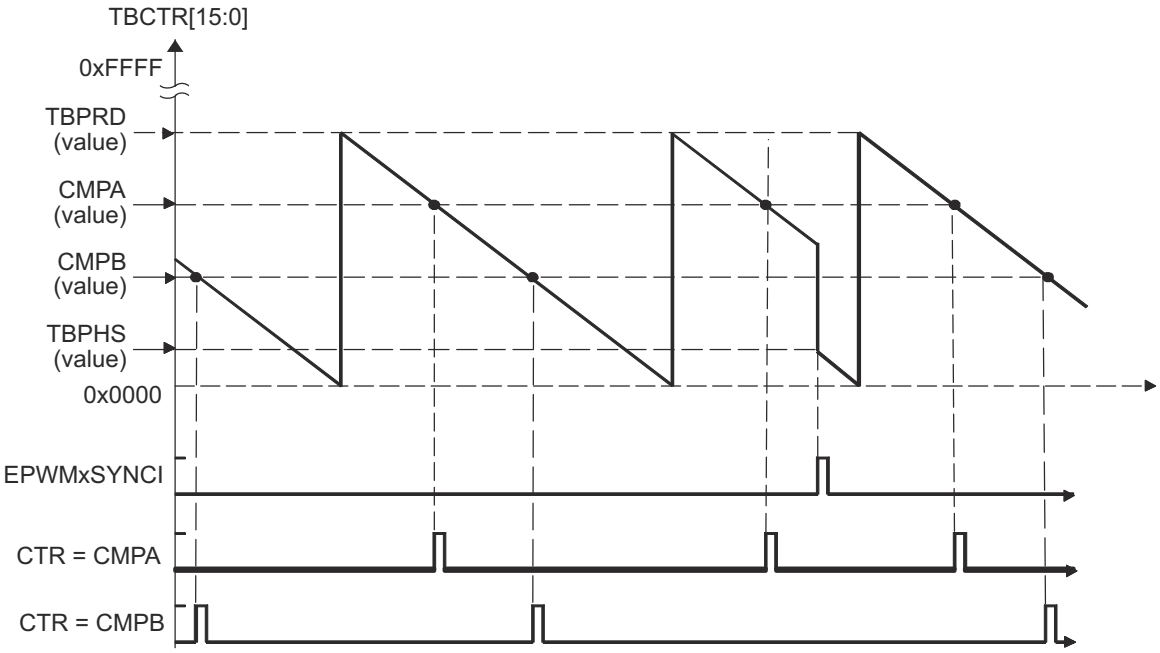


Figure 30-18. Counter-Compare Events in Down-Count Mode

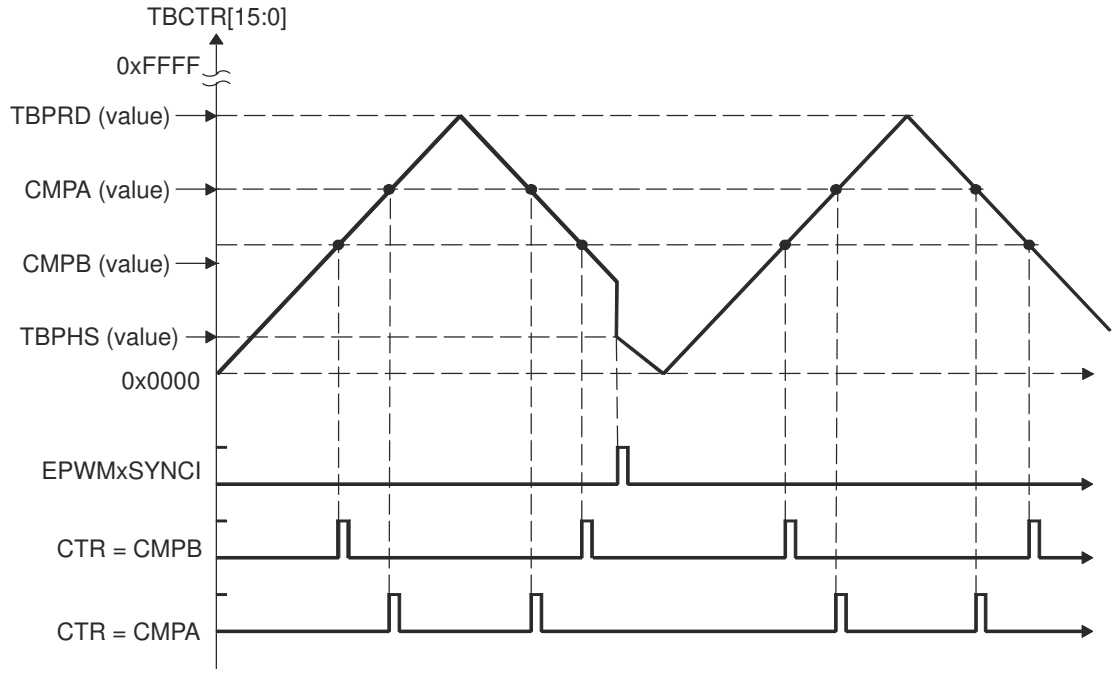


Figure 30-19. Counter-Compare Events In Up-Down-Count Mode, TBCTL[PHSDIR = 0] Count Down On Synchronization Event

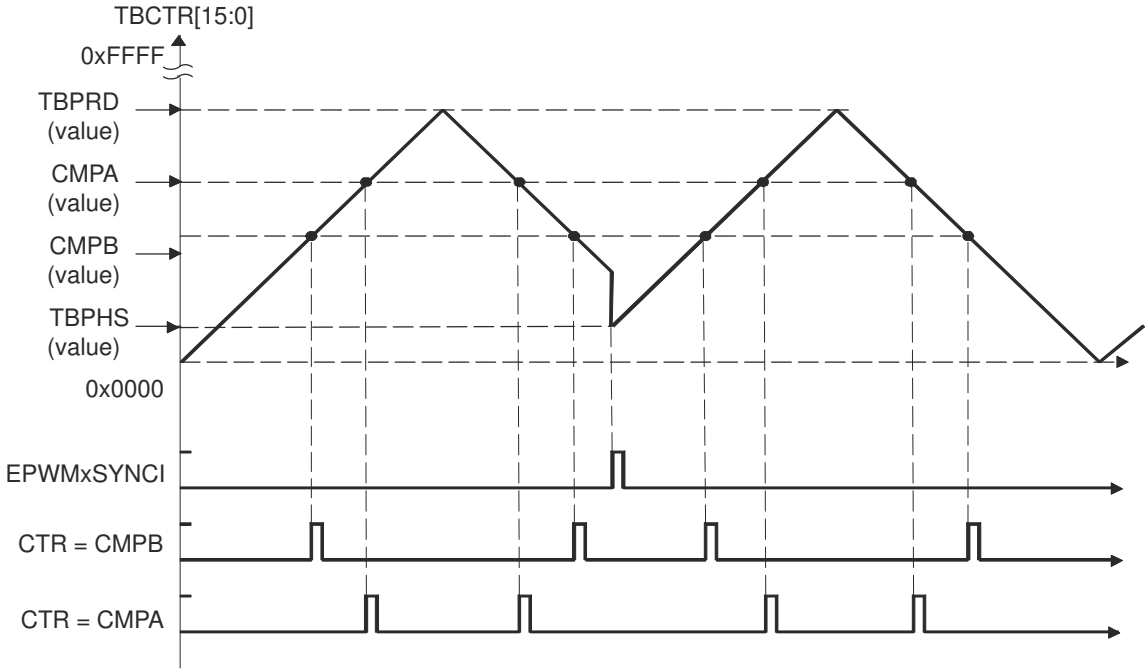


Figure 30-20. Counter-Compare Events In Up-Down-Count Mode, TBCTL[PHSDIR = 1] Count Up On Synchronization Event

30.6 Action-Qualifier (AQ) Submodule

The action-qualifier submodule has the most important role in waveform construction and PWM generation. The action-qualifier submodule decides which events are converted into various action types, thereby, producing the required switched waveforms at the EPWMxA and EPWMxB outputs.

Figure 30-21 illustrates the action-qualifier submodule within the ePWM.

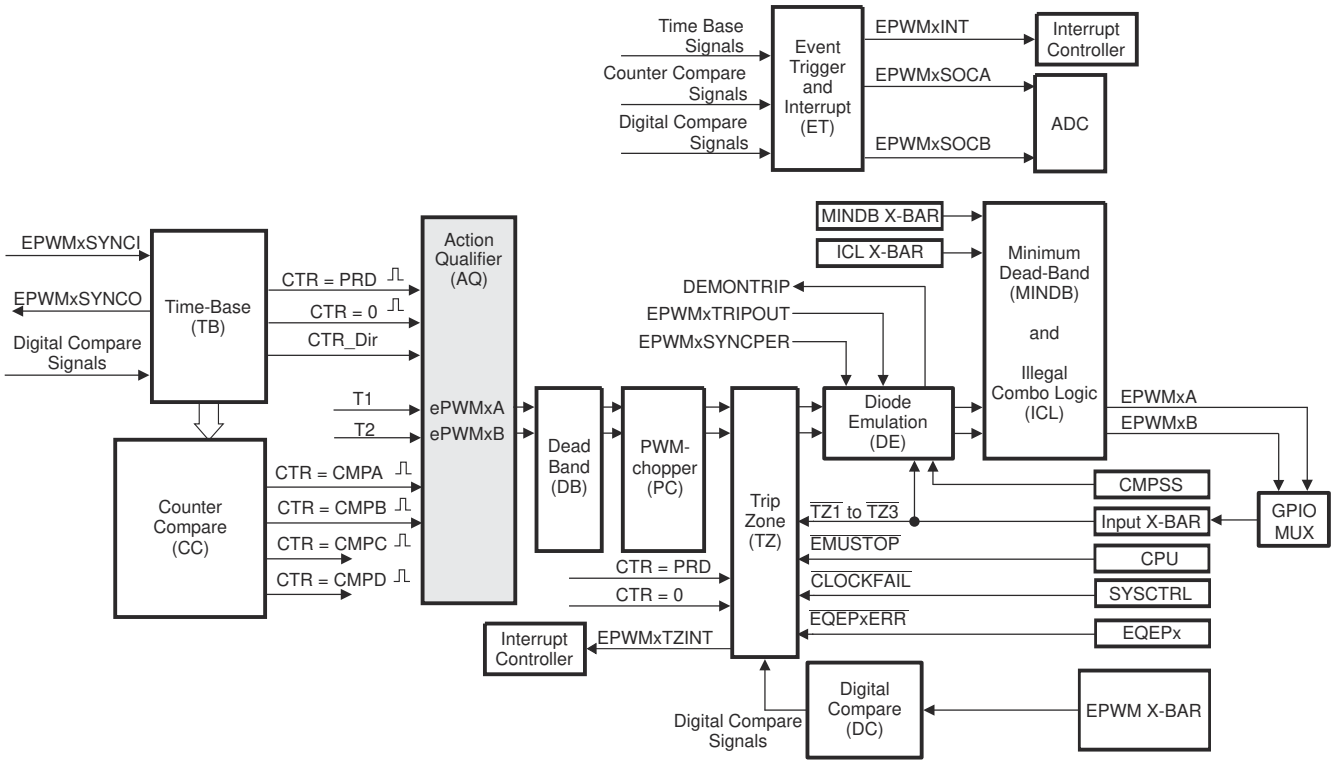


Figure 30-21. Action-Qualifier Submodule

30.6.1 Purpose of the Action-Qualifier Submodule

The action-qualifier submodule is responsible for the following:

- Qualifying and generating actions (set, clear, toggle) based on the following events:
  - CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD).
  - CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00)
  - CTR = CMPA: Time-base counter equal to the counter-compare A register (TBCTR = CMPA)
  - CTR = CMPB: Time-base counter equal to the counter-compare B register (TBCTR = CMPB)
- T1, T2 events: Trigger events based on comparator, trip or syncin events
- Managing priority when these events occur concurrently
- Providing independent control of events when the time-base counter is increasing and when the time-base counter is decreasing

30.6.2 Action-Qualifier Submodule Control and Status Register Definitions

The action-qualifier submodule operation is shown in Figure 30-22 and monitored by way of the registers in Section 30.20 .

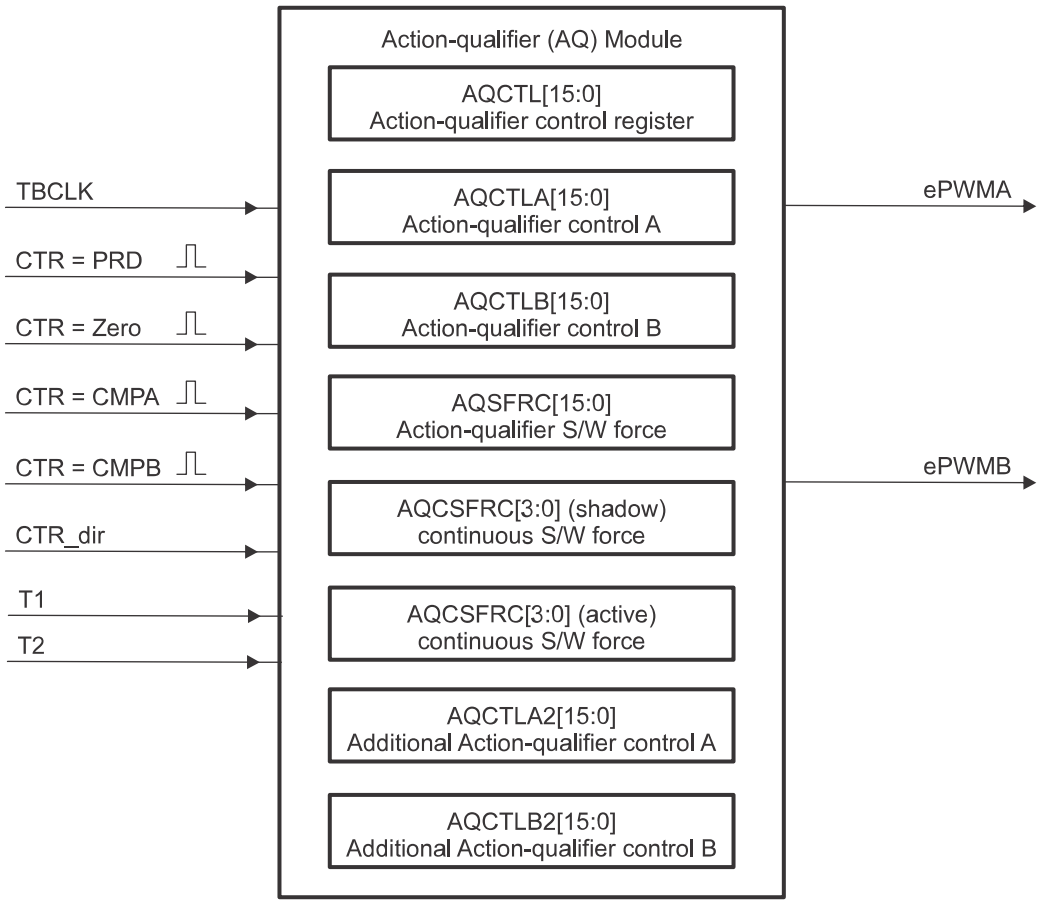


Figure 30-22. Action-Qualifier Submodule Inputs and Outputs

For convenience, the possible input events are summarized again in Table 30-4.

Table 30-4. Action-Qualifier Submodule Possible Input Events

Signal	Description	Registers Compared
CTR = PRD	Time-base counter equal to the period value	TBCTR = TBPRD
CTR = Zero	Time-base counter equal to 0	TBCTR = 0x00
CTR = CMPA	Time-base counter equal to the counter-compare A	TBCTR = CMPA
CTR = CMPB	Time-base counter equal to the counter-compare B	TBCTR = CMPB
T1 event	Based on comparator, trip, or syncin events	None
T2 event	Based on comparator, trip, or syncin events	None
Software forced event	Asynchronous event initiated by software	

The software forced action is a useful asynchronous event. This control is handled by the AQSFRC and AQCSFRC registers.

Note

If the CSFA is not used in shadow mode, the RLDCSF bit must be configured to disable shadow mode.

The action-qualifier submodule controls how the two outputs EPWMxA and EPWMxB behave when a particular event occurs. The event inputs to the action-qualifier submodule are further qualified by the counter direction (up or down). This allows for independent action on outputs on both the count-up and count-down phases.

The possible actions imposed on outputs EPWMxA and EPWMxB are:

- **Set High:** Set output EPWMxA or EPWMxB to a high level.
- **Clear Low:** Set output EPWMxA or EPWMxB to a low level.
- **Toggle:** If EPWMxA or EPWMxB is currently pulled high, then pull the output low. If EPWMxA or EPWMxB is currently pulled low, then pull the output high.
- **Do Nothing:** Keep outputs EPWMxA and EPWMxB at same level as currently set. Although the "Do Nothing" option prevents an event from causing an action on the EPWMxA and EPWMxB outputs, this event can still trigger interrupts and ADC start of conversion. See the description in Section 30.13 for details.

Actions are specified independently for either output (EPWMxA or EPWMxB). Any or all events can be configured to generate actions on a given output. For example, both CTR = CMPA and CTR = CMPB can operate on output EPWMxA. All qualifier actions are configured using the control registers found in the *ePWM Registers* section.

For clarity, the illustrations in this chapter use a set of symbolic actions. These symbols are summarized in Figure 30-23. Each symbol represents an action as a marker in time. Some actions are fixed in time (zero and period) while the CMPA and CMPB actions are moveable and the time positions are programmed by way of the counter-compare A and B registers, respectively. To turn off or disable an action, use the "Do Nothing option"(the default at reset).

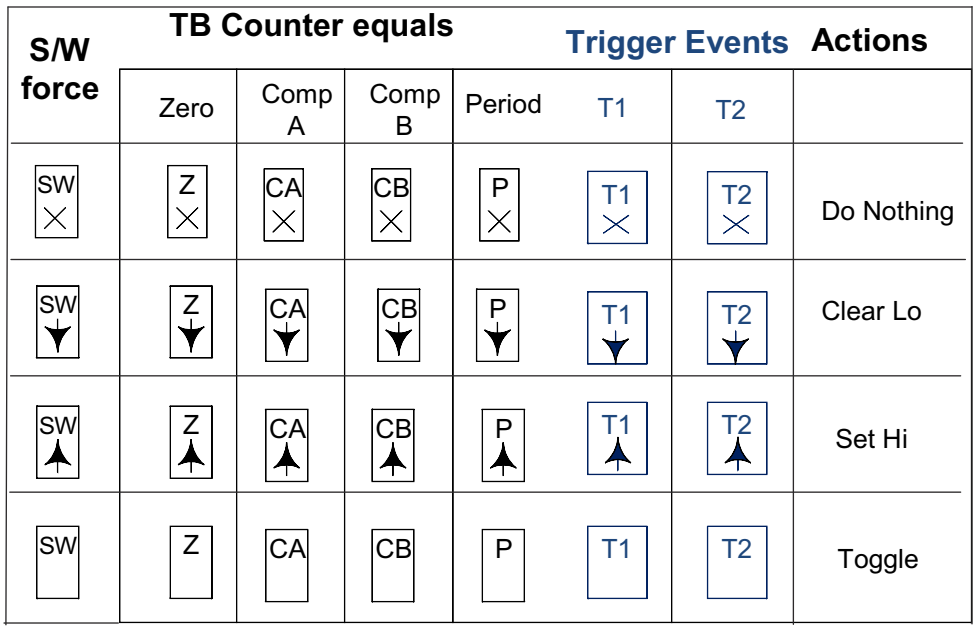


Figure 30-23. Possible Action-Qualifier Actions for EPWMxA and EPWMxB Outputs

The Action Qualifier Trigger Event Source Selection register (AQTSRCSEL) is used to select the source for T1 and T2 events. T1/T2 selection and configuration of a trip/digital-compare event in Action Qualifier submodule is independent of the configuration of that event in the Trip-Zone submodule. A particular trip event can or cannot

be configured to cause trip action in the Trip Zone submodule, but the same event can be used by the Action Qualifier to generate T1/T2 for controlling PWM generation.

30.6.3 Action-Qualifier Event Priority

It is possible for the ePWM action qualifier to receive more than one event at the same time. In this case, events are assigned a priority by the hardware. The general rule is events occurring later in time have a higher priority and software forced events always have the highest priority. The event priority levels for up-down count mode are shown in Table 30-5. A priority level of 1 is the highest priority and level 10 is the lowest. The priority changes slightly depending on the direction of TBCTR.

Table 30-5. Action-Qualifier Event Priority for Up-Down-Count Mode

Priority Level	Event If TBCTR is Incrementing TBCTR = Zero up to TBCTR = TBPRD	Event If TBCTR is Decrementing TBCTR = TBPRD down to TBCTR = 1
1 (Highest)	Software forced event	Software forced event
2	T1 on up-count (T1U)	T1 on down-count (T1D)
3	T2 on up-count (T2U)	T2 on down-count (T2D)
4	Counter equals CMPB on up-count (CBU)	Counter equals CMPB on down-count (CBD)
5	Counter equals CMPA on up-count (CAU)	Counter equals CMPA on down-count (CAD)
6 (Lowest)	Counter equals zero	Counter equals period (TBPRD)

Table 30-6 shows the action-qualifier priority for up-count mode. In this case, the counter direction is always defined as up; therefore, down-count events never are taken.

Table 30-6. Action-Qualifier Event Priority for Up-Count Mode

Priority Level	Event
1 (Highest)	Software forced event
2	Counter equal to period (TBPRD)
3	T1 on up-count (T1U)
4	T2 on up-count (T2U)
5	Counter equal to CMPB on up-count (CBU)
6	Counter equal to CMPA on up-count (CAU)
7 (Lowest)	Counter equal to Zero

Table 30-7 shows the action-qualifier priority for down-count mode. In this case, the counter direction is always defined as down; therefore, up-count events never are taken.

Table 30-7. Action-Qualifier Event Priority for Down-Count Mode

Priority Level	Event
1 (Highest)	Software forced event
2	Counter equal to Zero
3	T1 on down-count (T1D)
4	T2 on down-count (T2D)
5	Counter equal to CMPB on down-count (CBD)
6	Counter equal to CMPA on down-count (CAD)
7 (Lowest)	Counter equal to period (TBPRD)

It is possible to set the compare value greater than the period. In this case, the action takes place as shown in Table 30-8.

Table 30-8. Behavior if CMPA/CMPB is Greater than the Period

Counter Mode	Compare on Up-Count Event CAD/CBD	Compare on Down-Count Event CAD/CBD
Up-Count Mode	If CMPA/CMPB ≤ TBPRD period, then the event occurs on a compare match (TBCTR=CMPA or CMPB). If CMPA/CMPB > TBPRD, then the event does not occur.	Never occurs.
Down-Count Mode	Never occurs.	If CMPA/CMPB < TBPRD, the event occurs on a compare match (TBCTR=CMPA or CMPB). If CMPA/CMPB ≥ TBPRD, the event occurs on a period match (TBCTR=TBPRD).
Up-Down Count Mode	If CMPA/CMPB < TBPRD and the counter is incrementing, the event occurs on a compare match (TBCTR=CMPA or CMPB). If CMPA/CMPB is ≥ TBPRD, the event occurs on a period match (TBCTR = TBPRD).	If CMPA/CMPB < TBPRD and the counter is decrementing, the event occurs on a compare match (TBCTR=CMPA or CMPB). If CMPA/CMPB ≥ TBPRD, the event occurs on a period match (TBCTR=TBPRD).

30.6.4 AQCTLA and AQCTLB Shadow Mode Operations

To enable Action Qualifier mode changes which must occur at the end of a period even when the phase changes, shadowing of the AQCTLA and AQCTLB registers has been added on ePWMs type 2 and later. Additionally, shadow to active load on SYNC of these registers is supported as well. Shadowing of this register is enabled and disabled by the AQCTL[SHDWAQAMODE] and AQCTL[SHDWAQBMODE] bits. These bits enable and disable the AQCTLA shadow register and AQCTLB shadow register, respectively. The behavior of the two load modes is:

Shadow Mode:

The shadow mode for the AQCTLA is enabled by setting the AQCTL[SHDWAQAMODE] bit, and the shadow register for AQCTLB is enabled by setting the AQCTL[SHDWAQBMODE] bit. Shadow mode is disabled by default for both AQCTLA and AQCTLB

If the shadow register is enabled, then the content of the shadow register is transferred to the active register on one of the following events as specified by the AQCTL[LDAQAMODE], AQCTL[LDAQBMODE], AQCTL[LDAQASYNC], and AQCTL[LDAQBSYNC] register bits:

- CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD).
- CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00)
- Both CTR = PRD and CTR = Zero
- SYNC event caused by DCAEVT1 or DCBEVT1 or EPWMxSYNCI or TBCTL[SWFSYNC]
- Both SYNC event or a selection made by LDAQAMODE/LDAQBMODE

Global Load Support

Global load control mechanism can also be used for AQCTLA:AQCTLA2, AQCTLB:AQCTLB2, and AQCSFRC registers by configuring the appropriate bits in the global load configuration register (GLDCFG). When global load mode is selected, the transfer of contents from shadow register to active register for all registers that have this mode enabled, occurs at the same event as defined by the configuration bits in the Global Shadow to Active Load Control Register (GLDCTL). The global load control mechanism is explained in Section 30.4.7.

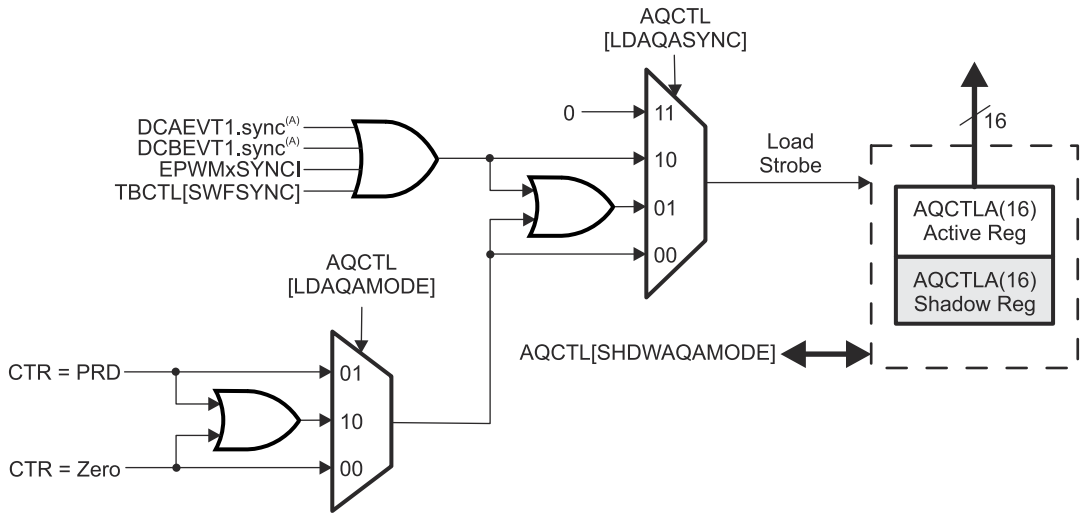
Immediate Load Mode:

If the immediate load mode is selected (that is, AQCTL[SHDWAQAMODE] = 0 or AQCTL[SHDWAQBMODE] = 0), then a read from or a write to the register goes directly to the active register. See Figure 30-24 and Figure 30-25.

Note

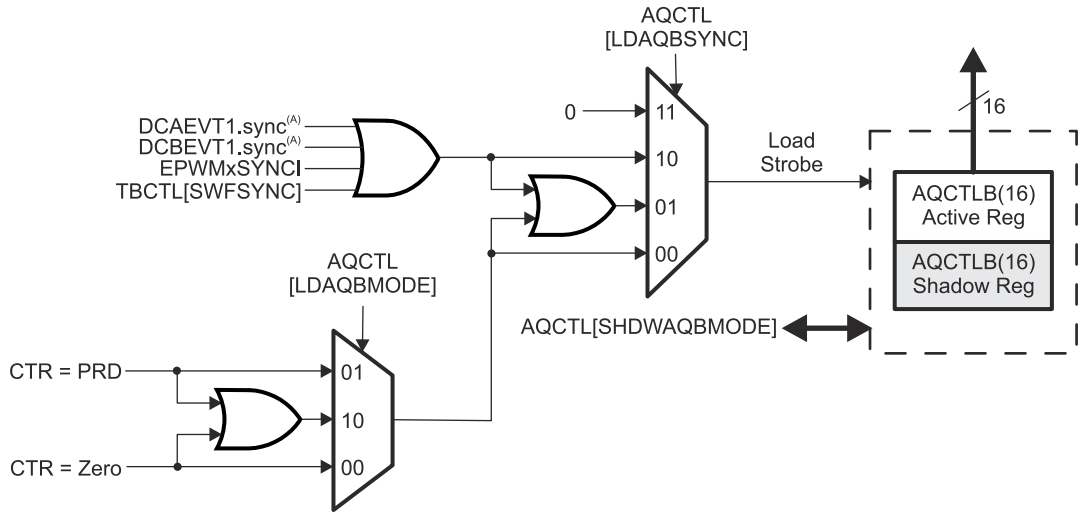
Shadow to Active Load of Action Qualifier Output A/B Control Register [AQCTLA and AQCTLB] on CMPA = 0 or CMPB = 0 boundary

If the Counter-Compare A Register (CMPA) or Counter-Compare B Register (CMPB) is set to a value of 0 and the action qualifier action on AQCTLA and AQCTLB is configured to occur in the same instant as a shadow to active load (that is, CMPA = 0 and AQCTLA shadow to active load on TBCTR = 0 using AQCTL register LDAQAMODE and LDAQAMODE bits), then both events enter contention. It is recommended to use a Non-Zero Counter-Compare when using Shadow to Active Load of Action Qualifier Output A/B Control Register on TBCTR = 0 boundary.



A. These events are generated by the ePWM digital compare (DC) submodule based on the levels of the TRIPIN inputs (for example, CMPSSx and $\overline{TZ}$ signals).

Figure 30-24. AQCTL[SHDWAQAMODE]



A. These events are generated by the ePWM digital compare (DC) submodule based on the levels of the TRIPIN inputs (for example, CMPSSx and $\overline{TZ}$ signals).

Figure 30-25. AQCTL[SHDWAQBMODE]

30.6.5 Configuration Requirements for Common Waveforms

Note

The waveforms in this chapter show the behavior of the ePWMs for a static compare register value. In a running system, the active compare registers (CMPA and CMPB) are typically updated from the respective shadow registers once every period. Specify when the update takes place: either when the time-base counter reaches zero or when the time-base counter reaches the period. There are some cases when the action based on the new value can be delayed by one period or the action based on the old value can take effect for an extra period. Some PWM configurations avoid this situation. These include, but are not limited to, the following:

Use up-down count mode to generate a symmetric PWM:

- If loading CMPA/CMPB on zero, then use CMPA/CMPB values greater than or equal to 1.
- If loading CMPA/CMPB on period, then use CMPA/CMPB values less than or equal to TBPRD-1.

  This means there is always a pulse of at least one TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

Use up-down count mode to generate an asymmetric PWM:

- To achieve 50%-0% asymmetric PWM use the following configuration: Load CMPA/CMPB on period and use the period action to clear the PWM and a compare-up action to set the PWM. Modulate the compare value from 0 to TBPRD to achieve 50%-0% PWM duty.

When using up-count mode to generate an asymmetric PWM:

- To achieve 0-100% asymmetric PWM, you **must** load CMPA/CMPB on TBPRD. When CMPA/CMPB is not loaded on TBCTR=PRD, boundary conditions can occur depending on the timing of the write and the value written to CMPA/CMPB. Use the Zero action to set the PWM and a compare-up action to clear the PWM. Modulate the compare value from 0 to TBPRD+1 to achieve 0-100% PWM duty.

When using up-count mode to generate an asymmetric PWM with deadband enabled:

- To achieve 0%-100% PWM use the following configuration: When the CMPA value is too close to 0 or PRD such that the following conditions are met (CMPX < Deadband) or (CMPX > PRD – Deadband), the actions specified by the AQCTL register for CMPX do not take effect. To avoid this, the AQCTL settings must be altered under these conditions only to generate either high or low pulses for both CAU or CAD events (both set or both clear). Make sure that this software update is occurring synchronous to the PWM carrier cycle, and shadow mode is enabled.

When using up-down count mode to generate an asymmetric PWM with deadband enabled:

- To achieve 0%-100% PWM use the following configuration: When the CMPA value is too close to 0 or PRD such that the following conditions are met (CMPX < Deadband/2) or (CMPX > PRD – (Deadband)/2), the actions specified by the AQCTL register for CMPX do not take effect. To avoid this, the AQCTL settings must be altered under these conditions only to generate either high or low pulses for both CAU or CAD events (both set or both clear). Make sure that this software update is occurring synchronous to the PWM carrier cycle, and shadow mode is enabled.

See *Using Enhanced Pulse Width Modulator (ePWM) Module for 0-100% Duty Cycle Control*.

Figure 30-26 shows how a symmetric PWM waveform can be generated using the up-down-count mode of the TBCTR. In this mode, 0%-100% DC modulation is achieved by using equal compare matches on the up count and down count portions of the waveform. In the example shown, CMPA is used to make the comparison. When the counter is incrementing, the CMPA match pulls the PWM output high. Likewise when the counter is decrementing, the compare match pulls the PWM signal low. When CMPA = 0, the PWM signal is high for the entire period giving a 100% duty waveform. When CMPA = TBPRD, the PWM signal is low achieving 0% duty.

When using this configuration in practice, if loading CMPA/CMPB on zero, then use CMPA/CMPB values greater than or equal to 1. If loading CMPA/CMPB on period, then use CMPA/CMPB values less than or equal to TBPRD-1. This means there is always a pulse of at least one TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

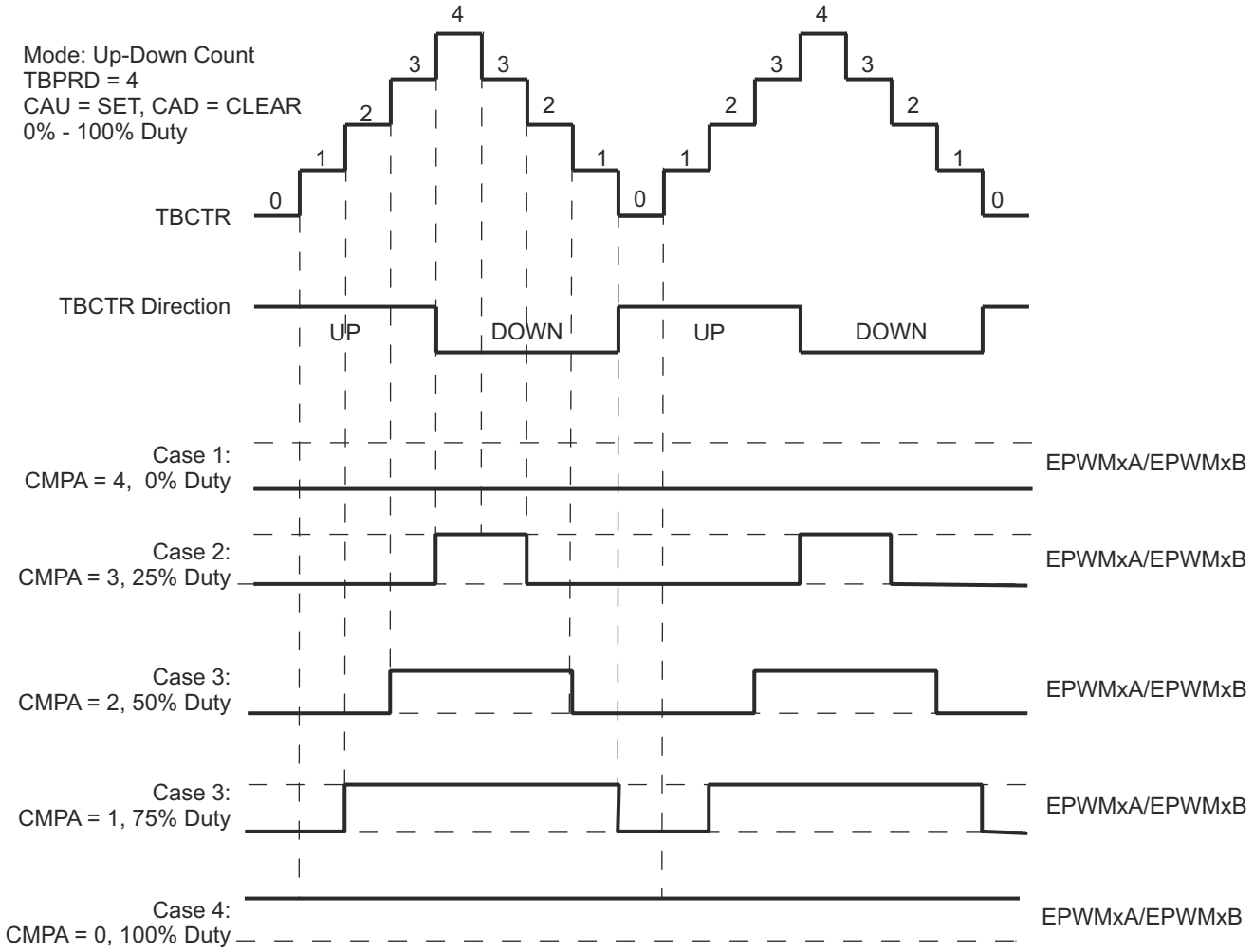
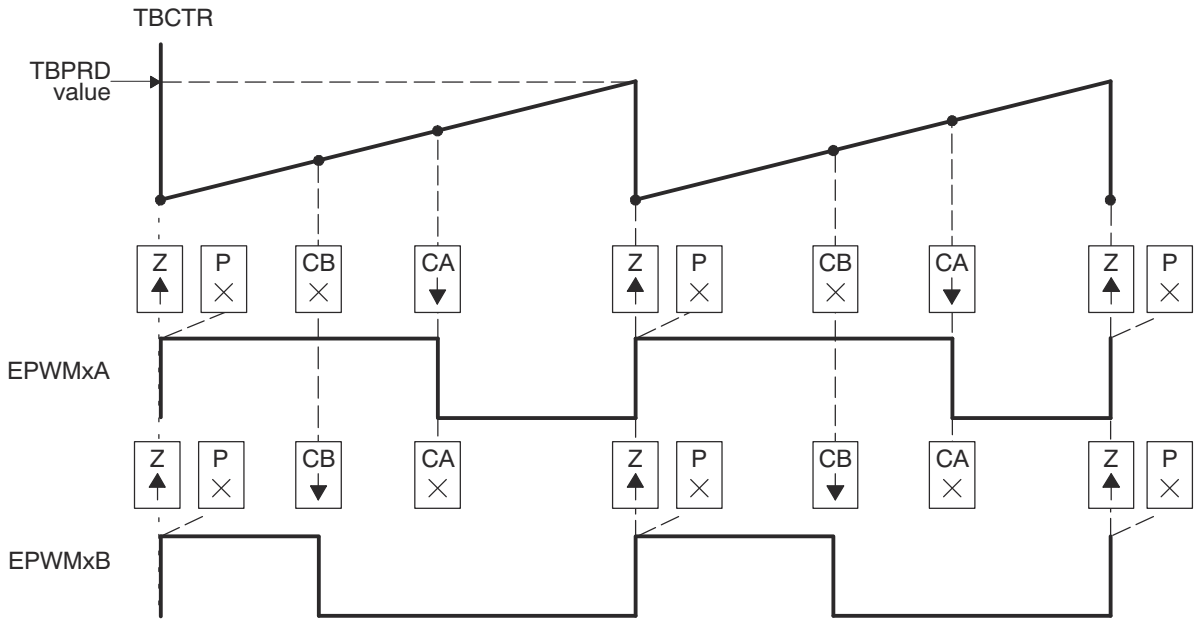


Figure 30-26. Up-Down Count Mode Symmetrical Waveform

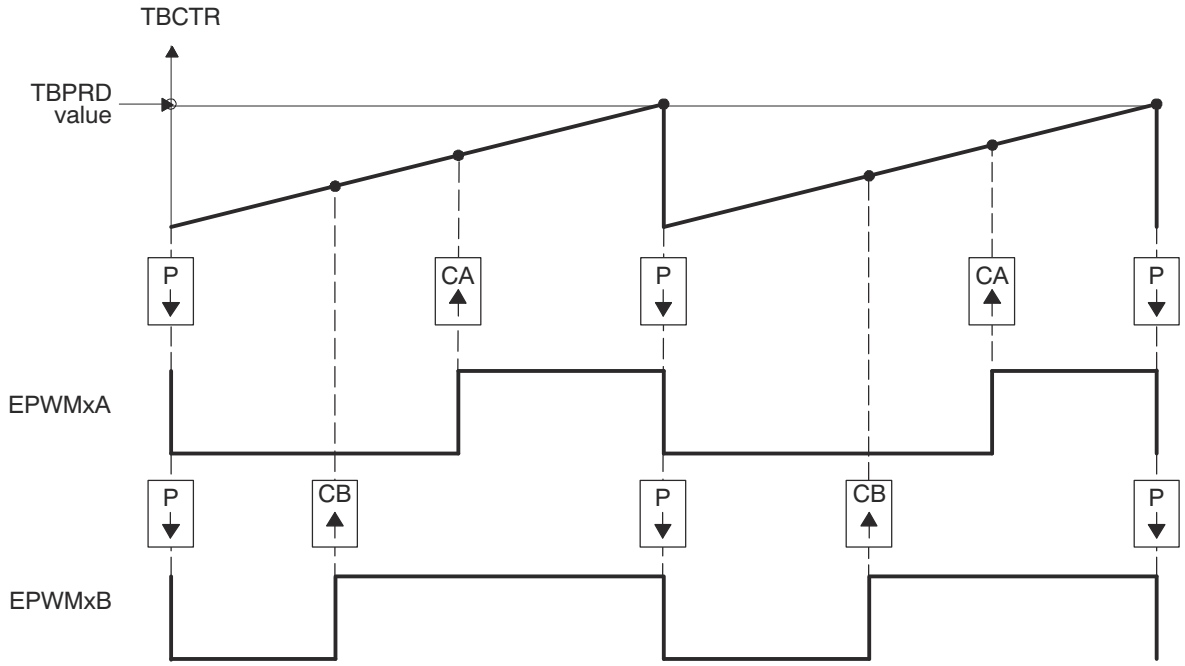
The PWM waveforms in Figure 30-27 through Figure 30-32 show some common action-qualifier configurations. Some conventions used in the figures and examples are as follows:

- TBPRD, CMPA, and CMPB refer to the value written in the respective registers. The active register, not the shadow register, is used by the hardware.
- CMPx, refers to either CMPA or CMPB.
- EPWMxA and EPWMxB refer to the output signals from ePWMx
- Up-Down means count-up-and count-down mode, Up means up-count mode and Down means down-count mode
- Sym = Symmetric, Asym = Asymmetric



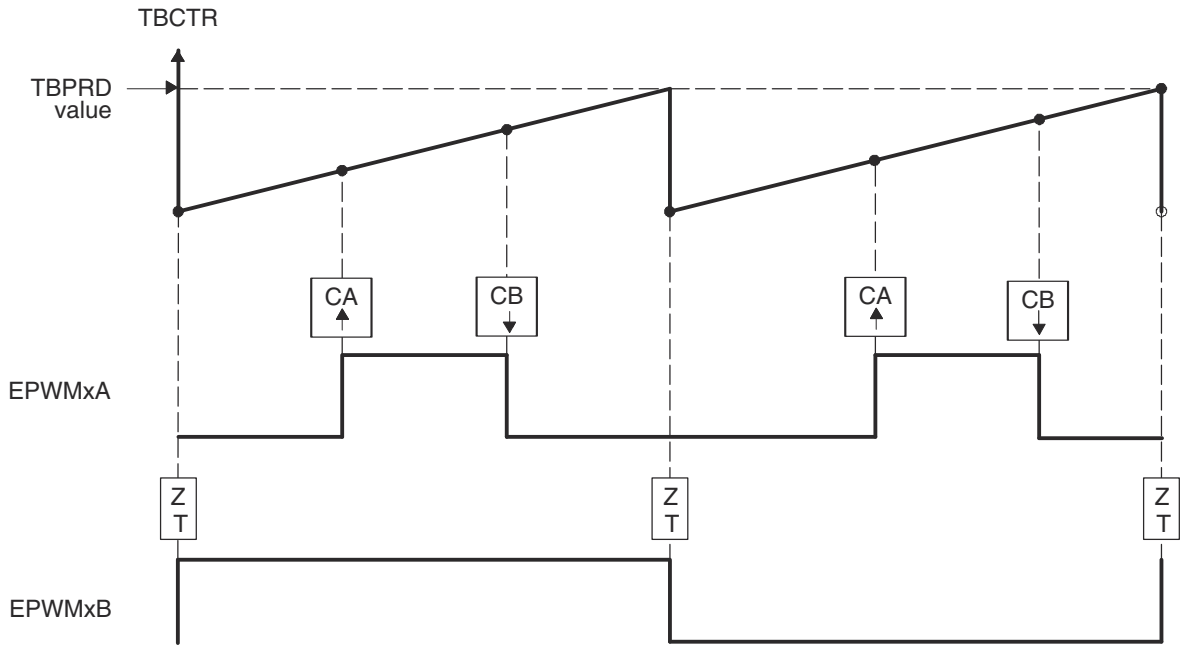
A. PWM period = (TBPRD + 1) × T_{TBCLK}
B. Duty modulation for EPWMxA is set by CMPA, and is active high (that is, high time duty proportional to CMPA).
C. Duty modulation for EPWMxB is set by CMPB and is active high (that is, high time duty proportional to CMPB).
D. The "Do Nothing" actions (X) are shown for completeness, but are not shown on subsequent diagrams.
E. Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCTR wraps from period to 0000.

Figure 30-27. Up, Single Edge Asymmetric Waveform, with Independent Modulation on EPWMxA and EPWMxB—Active High



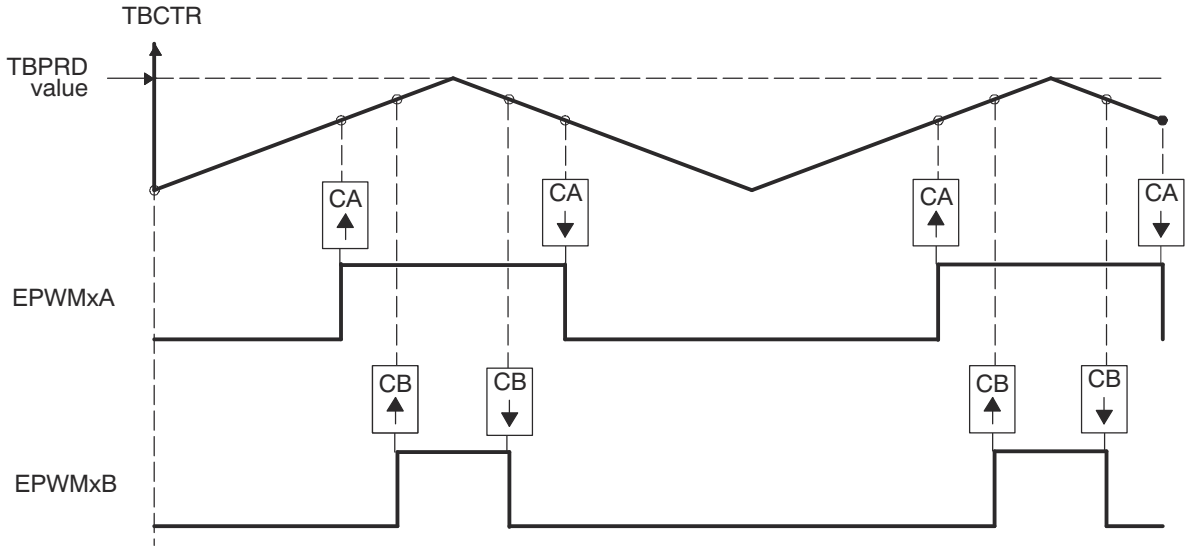
A. PWM period = (TBPRD + 1) × T_{TBCLK}
B. Duty modulation for EPWMxA is set by CMPA, and is active low (that is, the low time duty is proportional to CMPA).
C. Duty modulation for EPWMxB is set by CMPB and is active low (that is, the low time duty is proportional to CMPB).
D. Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCTR wraps from period to 0000.

Figure 30-28. Up, Single Edge Asymmetric Waveform with Independent Modulation on EPWMxA and EPWMxB—Active Low



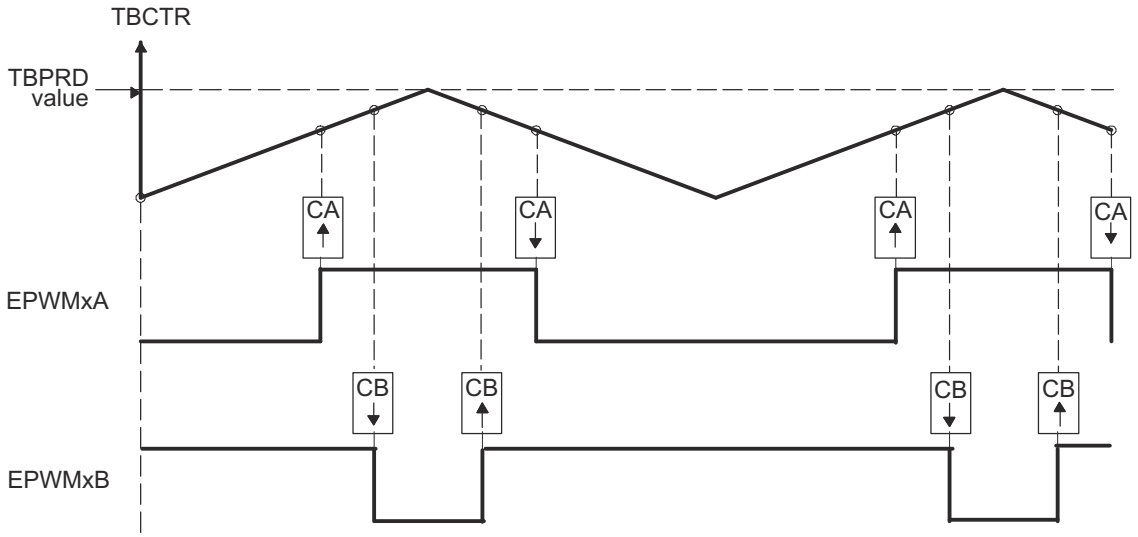
A. PWM frequency = 1/((TBPRD + 1) × T_{TBCLK})
B. Pulse can be placed anywhere within the PWM cycle (0000 - TBPRD)
C. High time duty proportional to (CMPB - CMPA)

Figure 30-29. Up-Count, Pulse Placement Asymmetric Waveform With Independent Modulation on EPWMxA



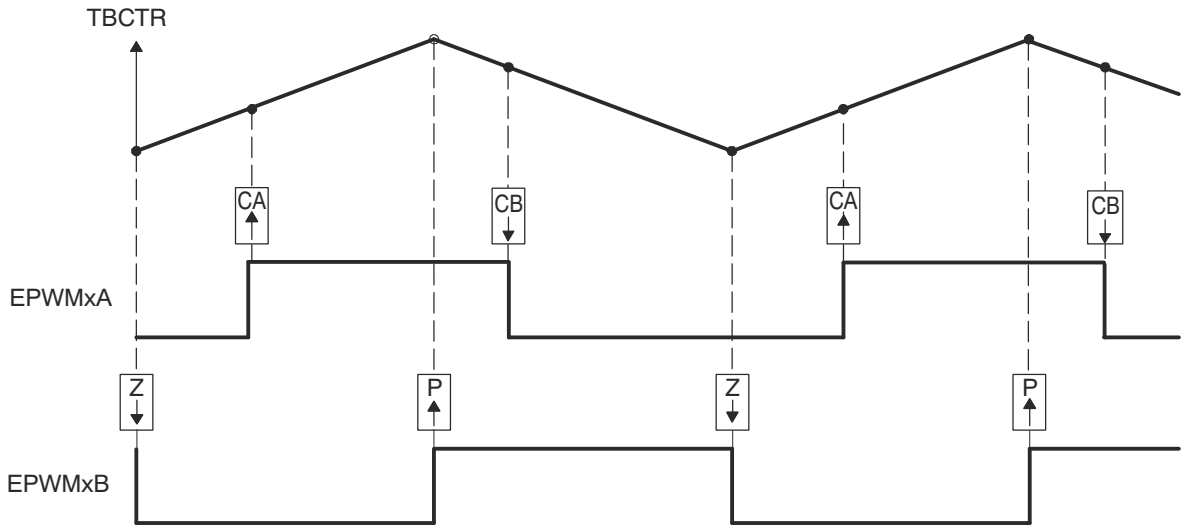
A. PWM period = 2 x TBPRD × T_{TBCLK}
B. Duty modulation for EPWMxA is set by CMPA, and is active low (that is, the low time duty is proportional to CMPA).
C. Duty modulation for EPWMxB is set by CMPB and is active low (that is, the low time duty is proportional to CMPB).
D. Outputs EPWMxA and EPWMxB can drive independent power switches.

Figure 30-30. Up-Down Count, Dual-Edge Symmetric Waveform, with Independent Modulation on EPWMxA and EPWMxB — Active Low



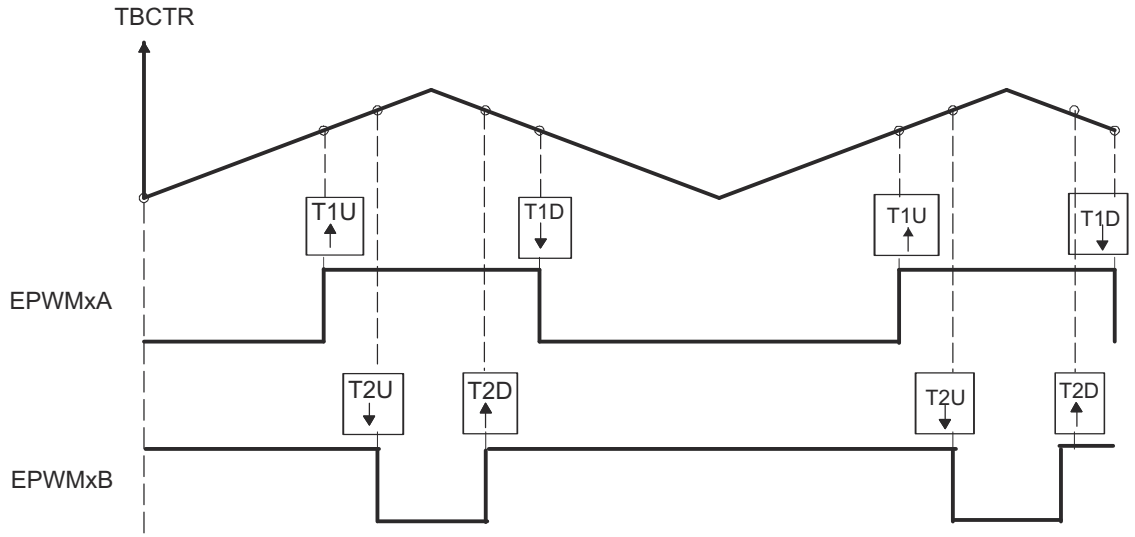
A. PWM period = 2 × TBPRD × T_{TBCLK}
B. Duty modulation for EPWMxA is set by CMPA, and is active low, that is, low time duty proportional to CMPA.
C. Duty modulation for EPWMxB is set by CMPB and is active high, that is, high time duty proportional to CMPB.
D. Outputs EPWMx can drive upper/lower (complementary) power switches.
E. Dead-band = CMPB - CMPA (fully programmable edge placement by software). Note the dead-band module is also available if the more classical edge delay method is required.

Figure 30-31. Up-Down Count, Dual-Edge Symmetric Waveform, with Independent Modulation on EPWMxA and EPWMxB — Complementary



A. PWM period = 2 × TBPRD × TBCLK
B. Rising edge and falling edge can be asymmetrically positioned within a PWM cycle. This allows for pulse placement techniques.
C. Duty modulation for EPWMxA is set by CMPA and CMPB.
D. Low time duty for EPWMxA is proportional to (CMPA + CMPB).
E. To change this example to active high, CMPA and CMPB actions need to be inverted (that is, Clear on CMPA, Set on CMPB).
F. Duty modulation for EPWMxB is fixed at 50% (utilizes spare action resources for EPWMxB).

Figure 30-32. Up-Down Count, Dual-Edge Asymmetric Waveform, with Independent Modulation on EPWMxA—Active Low



A. PWM period = 2 × TBPRD × TTBCLK
B. Independent T1 event actions when counter is counting up and when the counter is counting down are used to generate EPWMxA output.
C. Independent T2 event actions when counter is counting up and when the counter is counting down are used to generate EPWMxB output.
D. TZ1 is selected as the source for T1.
E. TZ2 is selected as the source for T2.

Figure 30-33. Up-Down Count, PWM Waveform Generation Utilizing T1 and T2 Events

30.7 XCMP Complex Waveform Generator Mode

The XCMP complex waveform generator mode is available in the type 5 ePWM and is enabled when XCMPEN is set. The main feature of the XCMP mode is to generate multiple ePWM pulses, with high resolution edge placement if needed, within one ePWM period.

XCMP features include:

- Up to eight counter compare registers XCMP1-XCMP8
- High resolution (HRPWM) edge placement support
- Up-Count counter mode support

Note

Down-Count and Up-Down-Count counter modes are not supported

- Pulse generation is only supported on XCMP1-8 matches (no support for counter events such as PRD and ZRO, or T1/T2 events)
- ePWM module synchronization is not allowed in XCMP mode

Note

The application software must disable the ePWM synchronization when XCMP mode is enabled.

- XCMP1-8 are loaded through CMPA and CMPB
- The eight XCMPn registers, can be allocated to either CMPA or CMPB through the application software configuration
- XAQCTLA and XAQCTLB registers determine the actions taken on the ePWM output for each XCMP1-8 counter matches
- Up to three ePWM period cycles can be configured at once through three shadow buffers

  - Each shadow buffer contains shadow registers for XCMP1-8, XTBPRD, XAQCTLA, XAQCTLB, CMPC, CMPD, and XMINMAX (which is used for CAPEVT signal generation)
  - Shadow buffer SHDW2 and SHDW3 can be repeated up to eight times
- All ePWM modules can be linked to trigger the start of the shadow loading at the same time through EPWMXLINKXLOAD

Note

Enabling and disabling XCMP mode during run time can cause unintended events to occur. The recommendation is to stay in XCMP mode if XCMP mode is enabled. If enabling and disabling XCMP mode during run time is required, a peripheral reset is required after disabling XCMP mode on the PWM module.

30.7.1 XCMP Allocation to CMPA and CMPB

The first criteria that must be selected is whether both EPWM channel A and channel B outputs are required. If both channel A and channel B are required, XCMP registers must be assigned to both CMPA and CMPB. The XCMPn registers loaded to CMPA are used for configuring the A channel through XAQCTLA actions. The XCMPn registers loaded to CMPB are used for configuring the B channel through XAQCTLB actions.

XCMP allocation to CMPA and CMPB is done through XCMPCTL1.XCMPSPLIT. If both channel A and channel B are required in the system, then the XCMPCTL1.XCMPSPLIT must be set. This allows CMPA to use XCMP1-n (where n has a maximum value of 4) while CMPB uses XCMP5-m (where m has a maximum value of 8). If only channel A is needed, then XCMPCTL1.XCMPSPLIT must be cleared, allowing CMPA to use XCMP1-n (where n has a maximum value of 8), which means up to eight edges can be generated on channel A.

Note

The maximum number of edges that channel B can have is four, when XCMP5-8 are allocated to CMPB and all four XCMP5-8 are used by setting the XCMPB_ALLOC to use all available XCMPs.

XCMPA_ALLOC and XCMPB_ALLOC determines how many of the available XCMPs for each CMPA and CMPB must be used in the ePWM configuration.

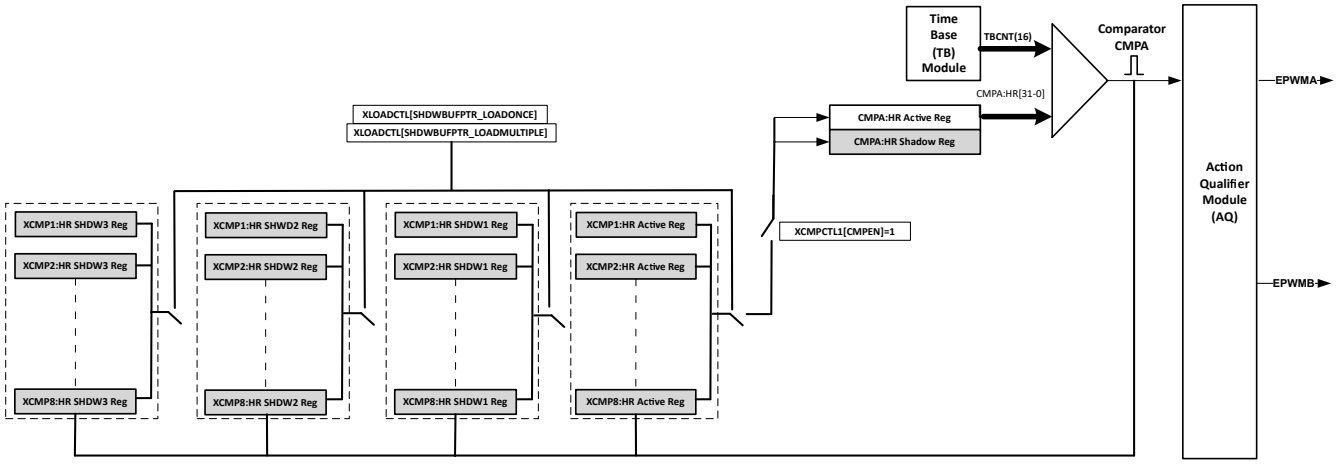


Figure 30-34. Allocate All XCMP1-8 to CMPA

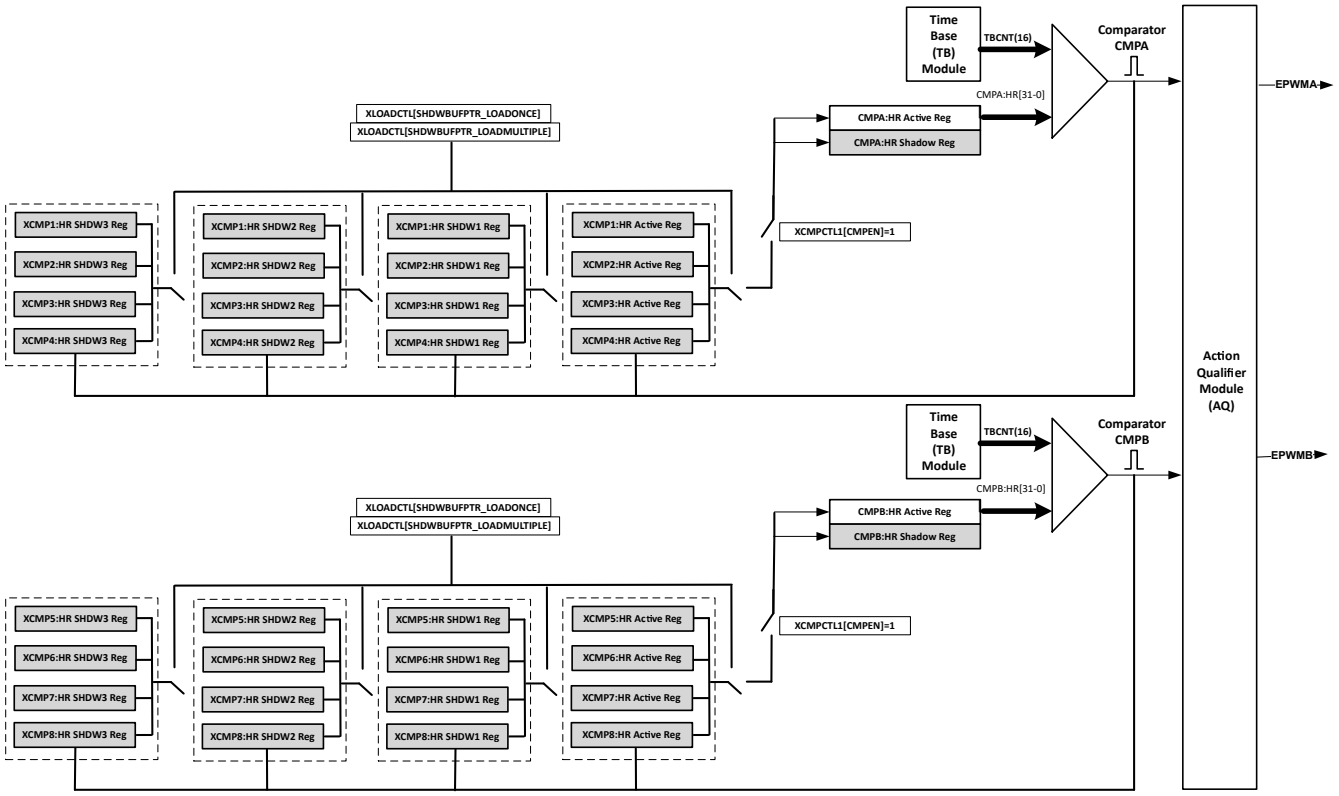


Figure 30-35. XCMP1-4 Allocated to CMPA and XCMP5-8 Allocated to CMPB

30.7.2 XCMP Shadow Buffers

Three SHDW buffers are available for XCMP configurations. Each SHDW buffer contains the XCMP1-8 values (CMPA and CMPB values), XTBPRD (TBPRD value), XCMPC (CMPC value), XCMPD (CMPD value), XAQCTLA and XAQCTLB. Each SHDW buffer also contains the XMINMAX values which are used for CAPEVT signal generation.

With the three SHDW buffer (SHDW1, SHDW2 and SHDW3) the values used for the upcoming ePWM period cycles can be buffered.

With XCMPEN set, the load of the active registers are controlled by the XLOADCTL and XLOAD registers. The shadow to active loading of the registers (other than XMINMAX, XCMPC, XCMPD) are always done three cycles prior to TBCTR==ZERO event. XMINMAX, XCMPC and XCMPD shadow loading is done at TBCTR==PRD.

There are two load modes configured by XLOADCTL[LOADMODE]:

- **LOADONCE** Mode (XLOADCTL[LOADMODE] = 0)
  - In LOADONCE mode, XLOADCTL[SHDWBUFPTR_ LOADONCE] is used to set the pointer location of the shadow buffer.
  - XLOADCTL[SHDWBUFPTR_ LOADONCE] is set by the user and is **NOT** automatically decremented. Upon the occurrence of the first load strobe (write of '1' to XLOAD[STARTLD] bit), active register set is loaded from the XLOADCTL[SHDWBUFPTR_ LOADONCE] SHDW selected by the user. Further load strobes are ignored, and ePWM waveform generation continues with the active register set until next XLOAD[STARTLD] is initiated.
  - When the software sets the XLOAD[STARTLD] bit again, the active register set is loaded from the XLOADCTL[SHDWBUFPTR_ LOADONCE] SHDW selected by the user. If the user wants to initiate a SHDW load from a different shadow register set, then the software can update the XLOADCTL[SHDWBUFPTR_ LOADONCE] register accordingly before setting the XLOADCTL[STARTLD].

- **LOADMULTIPLE** Mode (XLOADCTL[LOADMODE] = 1)
  - XLOADCTL[SHDWBUFPTR_ LOADMULTIPLE] always points to the current shadow register set that is loaded into the active registers set.
  - Setting the XLOAD[STARTLD] bit initiates a load strobe. The SHDW buffer pointer resets to XLOADCTL[SHDWLEVEL] and the corresponding buffer contents are loaded to the active register set. When the next valid load strobe arrives, XLOADCTL[SHDWBUFPTR_ LOADMULTIPLE] is decremented by 1 and the corresponding buffer contents are loaded to the active register set. This continues until the XLOADCTL[SHDWBUFPTR_ LOADMULTIPLE] value reaches 1. At this time SHDW1 values get copied to the active register set. Further load strobes are ignored and the ePWM waveform generation continues with the active register set until next XLOAD[STARTLD] is initiated.
  - Once the XLOADCTL[SHDWBUFPTR_ LOADMULTIPLE] value reaches 1, no further decrements to the this pointer are done until the next STARTLD initiation. This means the XLOADCTL[SHDWBUFPTR_ LOADMULTIPLE] remains at value of '1', indicating that the SHDW1 register set is in use till the next load initiation by user.
  - For a SHDWLEVEL of 3 buffers SHDW3 is loaded first followed by SHDW2 and SHDW1. Then until the next STARTLD write by the software, the SHDW1 values are in use.

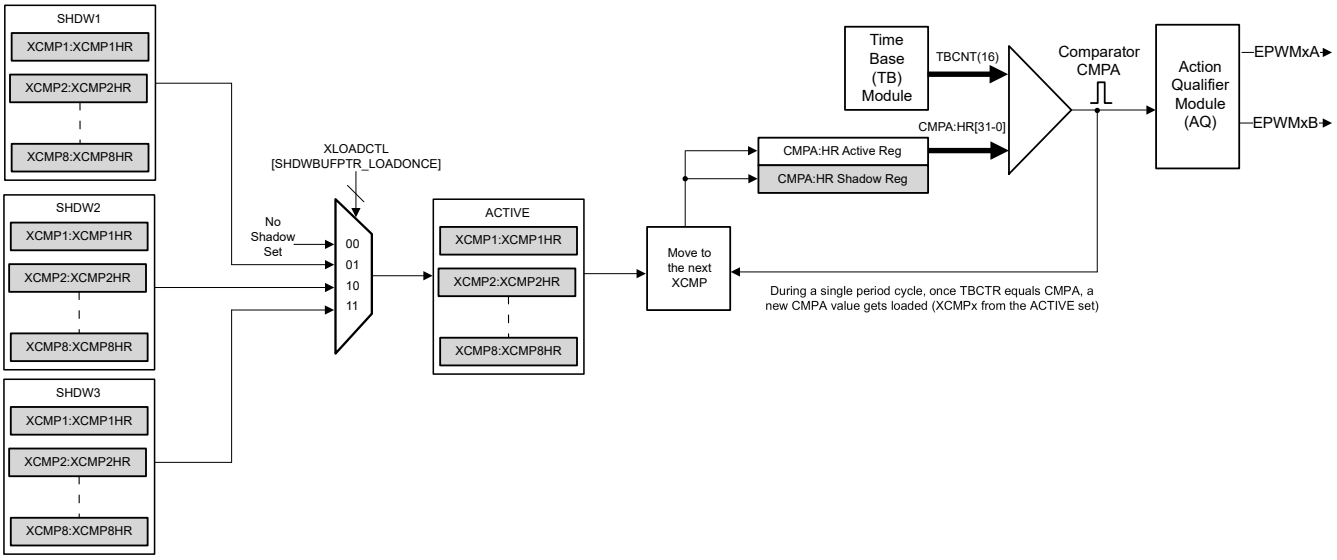


Figure 30-36. XCMP- Load Once Functionality

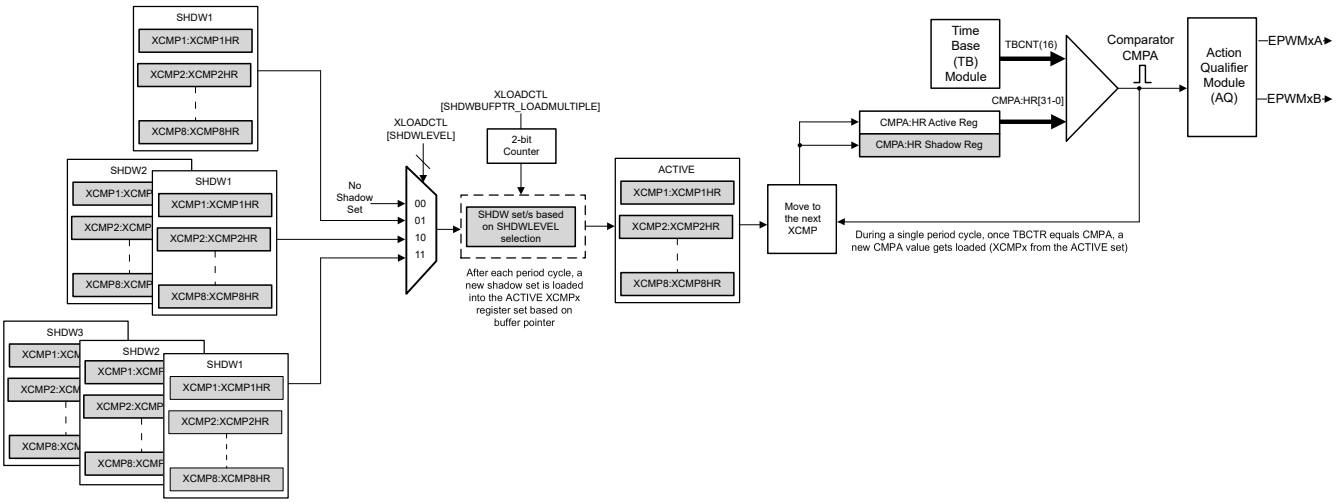


Figure 30-37. XCMP- Load Multiple Functionality

With this new loading scheme, the global load functionality also changes when using XCMP mode. In this new configuration, once a write to STARLD occurs, the next time the time base counter equals zero or a force load software write occurs, the shadow buffer pointers get reset, based on the load mode (load once or load multiple).

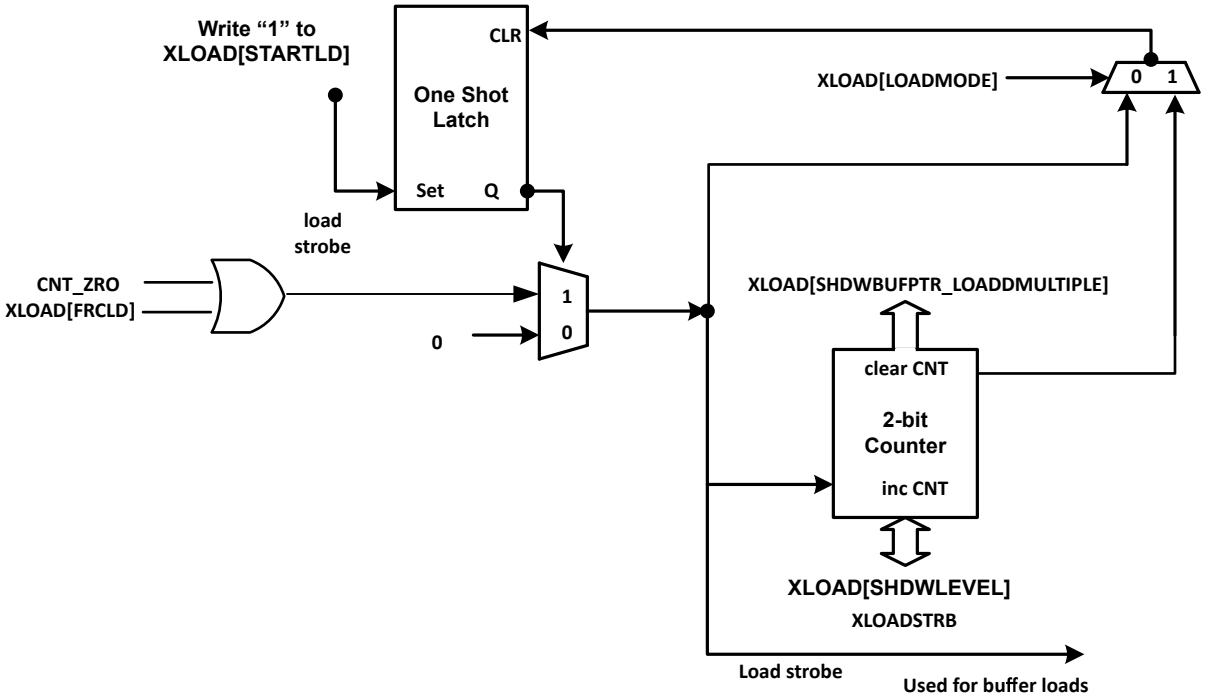


Figure 30-38. Global Load: Signals and Registers

Shadow buffers can also be repeated more than once. Shadow buffer repeat counters are:

- Users can optionally repeat each shadow buffer multiple times. This option sets the repeat count for SHDW2 and SHDW3 buffers before the pointer moves to the SHDW1 buffer. SHDW1 buffer by default repeats until the next load is initiated by the software and hence there is no configurable repeat option for SHDW1 buffer.
- Repeat counter option of the shadow buffers is applicable in LOADMULTIPLE mode. In the LOADONCE mode, user can manually keep track of the repeat counts and move to the SHDW pointer buffer.
- Each shadow buffer has a 3-bit counter. Each buffer can be set to repeat up to 8 times before moving the pointer to the next buffer.
- XLOADCTL[RPTBUF2PRD] and XLOADCTL[RPTBUF3PRD] are used to control the repeat period for each SHDW buffer.

No shadowing can be set by setting the XLOADCTL[SHDWLEVEL] to '0'. In this case, the ACTIVE registers are available for use (XCMP1_ACTIVE, XCMP2_ACTIVE, and so on).

30.7.3 XCMP Operation

The XCMP complex waveform generation mode is described in this section.

The XCMP mode can be used to generate multiple edges within one ePWM period. The application software must write the location of the ePWM waveform edges to the XCMP registers. Each XCMPn register assigned and used for an ePWM CMPx (CMPA or CMPB) must be spaced out according to the following guidelines to make sure of correct waveform generation.

Figure 30-39 shows an example of four XCMP values being loaded into CMPA during one period cycle and the remaining four XCMP values being used for CMPB. When the action for the last XCMP value loaded into CMPA/CMPB in a period is met, the last value for CMPA/CMPB remains until the next time TBCTR = 0 due to a new shadow set load.

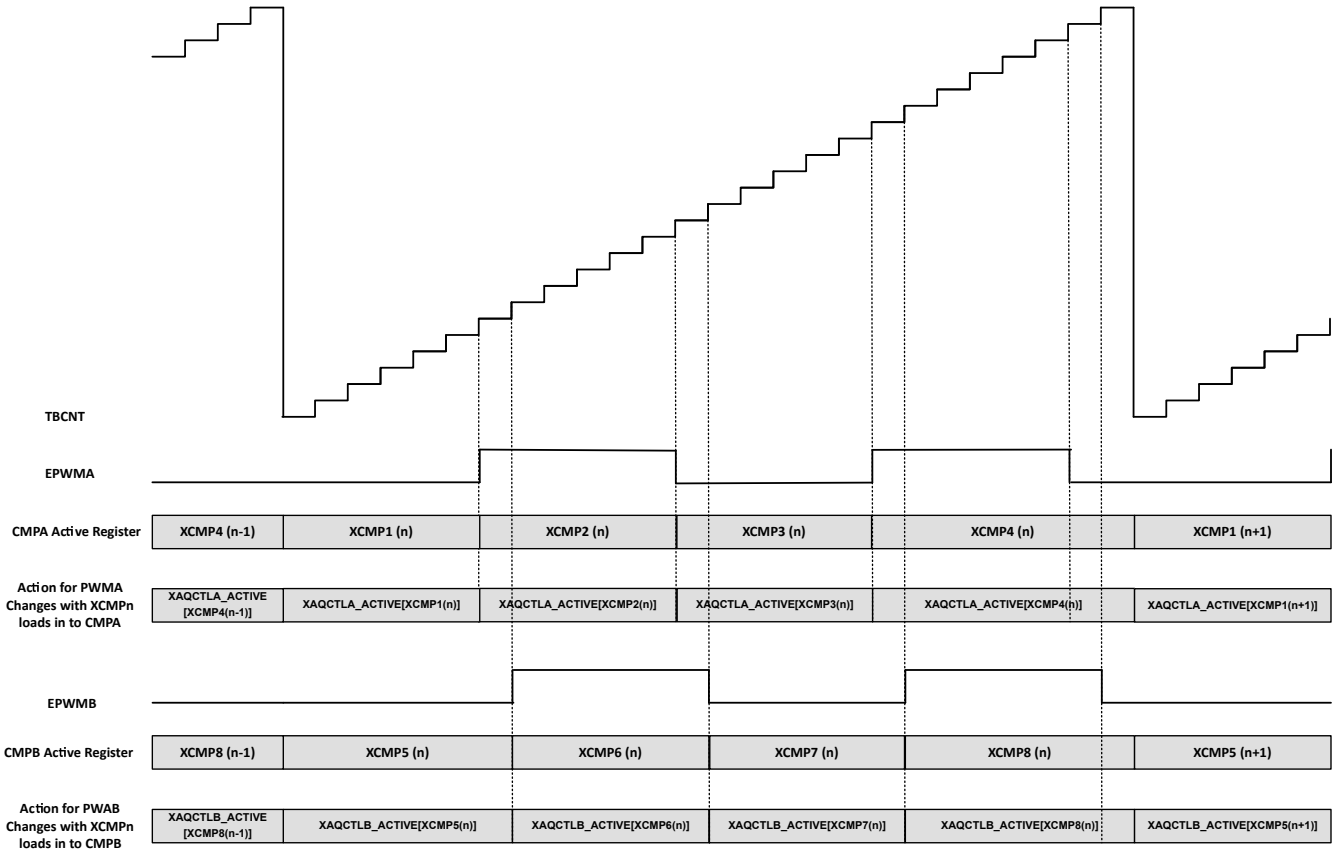


Figure 30-39. CMPA and CMPB values being loaded from XCMP registers

Assume XCMP1-3 are assigned and used by CMPA (XCMP4 is not used), and XCMP5-6 are assigned and used by CMPB (XCMP7 and XCMP8 are not used):

- For XCMP1-8 to be split between CMPA and CMPB, software must write XCMPCTL1[XCMPSPLIT] = 1
- For CMPA to only use XCMP1-3, software must write XCMPCTL1[CMPA_ALLOC] = 3
- For CMPB to only use XCMP5-6, software must write XCMPCTL1[CMPB_ALLOC] = 6

For XCMP1-3 in this scenario, since all are used by CMPA, the values written to XCMP1, XCMP2, and XCMP3 must:

- Without high-resolution edge placement requirement: XCMP(n+1) > (XCMPn) + 1
- With high-resolution edge placement requirement: XCMP(n+1) > (XCMPn) + 3

The requirements above for the minimum difference between XCMP(n+1) and XCMPn must be met in the application software.

The actions taken for each XCMP1-8 must be configured in XAQCTLA and XAQCTLB.

If shadowing is required then the XCMP1-8, XAQCTLA and XAQCTLB values must be written to the corresponding shadow buffer. As an example, Table 30-9 shows how the shadow buffers are used in LOADMULTIPLE mode.

The SHDW buffers 2 and 3 can also be repeated more than once by using the RPTBUF2PRD and RPTBUF3PRD.

Table 30-9. SHDW Buffer Loading Example

	XCMPn, XTBPRD			XTBPRD, TBPRD	XCMPn: XCMPnHR	CMPA: CMPAHR	What happens next?
	SHDW3FULL	SHDW2FULL	SHDW1FULL	Active	Active	Active	
CPU Initialization	Set	Set	Set				Registers initialized by CPU. Load event occurs.
ePWM Cycle 1	Clear	Set	Set	XTBPRD_SHDW3	XCMPn_SHDW3	XCMPn_SHDW3	SHDWBUFPTR set to 3
ePWM Cycle 2	Clear	Clear	Set	XTBPRD_SHDW2	XCMPn_SHDW2	XCMPn_SHDW2	SHDWBUFPTR set to 2
ePWM Cycle 3	Clear	Clear	Clear	XTBPRD_SHDW1	XCMPn_SHDW1	XCMPn_SHDW1	SHDWBUFPTR set to 1
ePWM Cycle 4	Clear	Clear	Clear	XTBPRD_SHDW1	XCMPn_SHDW1	XCMPn_SHDW1	SHDWBUFPTR set to 1 No shadow to active loading from buffer. Operation continues with values in XCMPn_ACTIVE registers.
ePWM Cycle 5	Clear	Clear	Clear	XTBPRD_SHDW1	XCMPn_SHDW1	XCMPn_SHDW1	SHDWBUFPTR set to 1 Continues operation with same values in XCMPn_ACTIVE until the next buffer load event
CPU Load (During ePWM Cycle 5)	Set	Set	Set	XTBPRD_SHDW1	XCMPn_SHDW1	XCMPn_SHDW1	CPU loads new shadow value set. Load event occurs. SHDWBUFPTR set to 3
ePWM Cycle 6	Clear	Set	Set	XTBPRD_SHDW3	XCMPn_SHDW3	XCMPn_SHDW3	SHDWBUFPTR set to 3
ePWM Cycle 7	Clear	Clear	Set	XTBPRD_SHDW2	XCMPn_SHDW2	XCMPn_SHDW2	SHDWBUFPTR set to 2
ePWM Cycle 8	Clear	Clear	Clear	XTBPRD_SHDW1	XCMPn_SHDW1	XCMPn_SHDW1	SHDWBUFPTR set to 1
ePWM Cycle 9	Clear	Clear	Clear	XTBPRD_SHDW1	XCMPn_SHDW1	XCMPn_SHDW1	Continues operation with the same values in XCMPn_ACTIVE until the next buffer load event. SHDWBUFPTR set to 1
ePWM Cycle 10	Set	Set	Set	XTBPRD_SHDW1	XCMPn_SHDW1	XCMPn_SHDW1	CPU loads new shadow register set. Load event occurs. SHDWBUFPTR set to 3

30.8 Dead-Band Generator (DB) Submodule

Figure 30-40 illustrates the dead-band submodule within the ePWM.

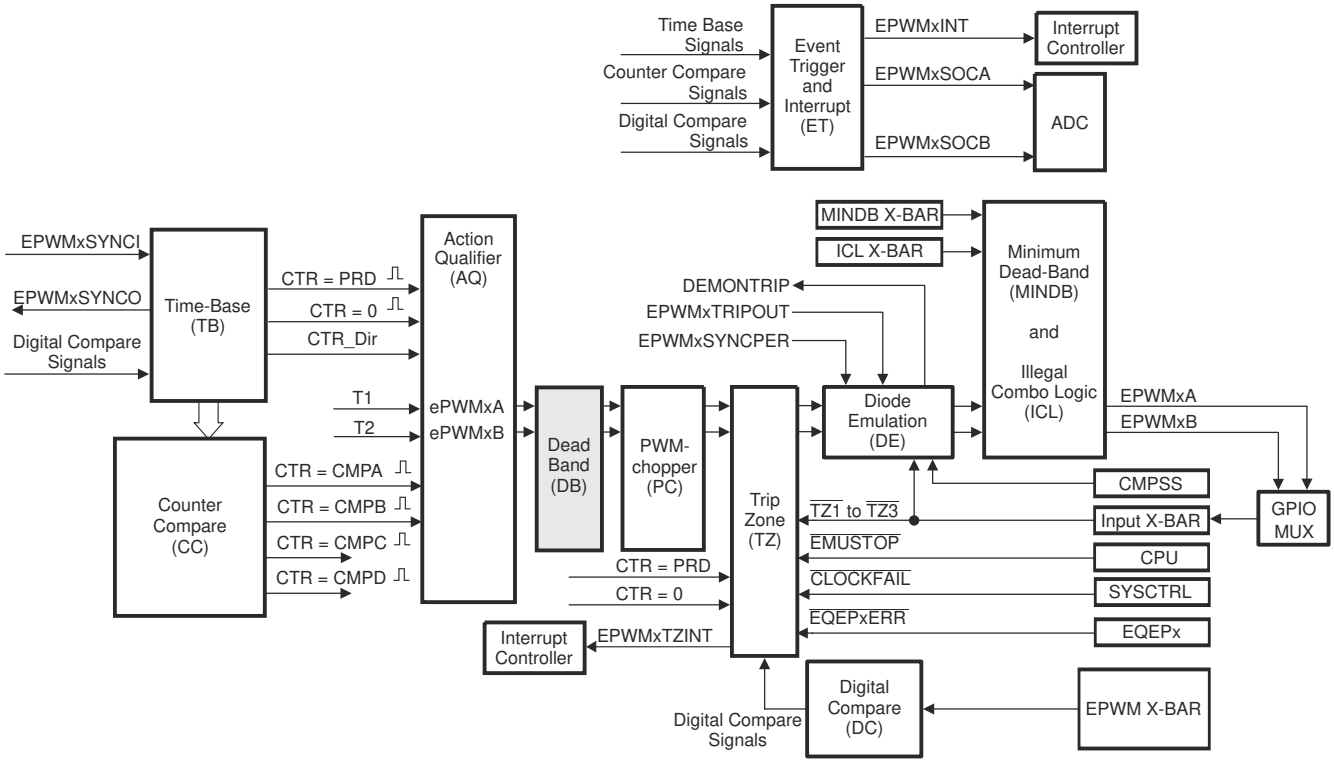


Figure 30-40. Dead_Band Submodule

30.8.1 Purpose of the Dead-Band Submodule

The action-qualifier (AQ) module section discussed how the AQ module can generate the required dead band by having full control over edge placement using both the CMPA and CMPB resources of the ePWM module. However, if the more classical edge delay-based dead band with polarity control is required, then the dead-band submodule described here must be used.

The key functions of the dead-band module are:

- Generating appropriate signal pairs (EPWMxA and EPWMxB) with dead-band relationship from a single EPWMxA input
- Programming signal pairs for:
  - Active high (AH)
  - Active low (AL)
  - Active high complementary (AHC)
  - Active low complementary (ALC)
- Adding programmable delay to rising edges (RED)
- Adding programmable delay to falling edges (FED)
- Can be totally bypassed from the signal path (note dotted lines in diagram)

30.8.2 Dead-band Submodule Additional Operating Modes

On type 1 ePWM RED can appear on one channel output and FED can appear on the other channel output.

The following list shows the distinct difference between type 1 and type 4 modules with respect to dead-band operating modes:

- By adding S6, S7, and S8 in Figure 30-41, RED and FED can appear on both the A-channel and B-channel outputs. Additionally, both RED and FED together can be applied to either the A-channel or B-channel outputs to allow B-channel phase shifting with respect to the A-channel.

Note

Phase shifting B-channel with respect to the A-channel using the dead-band submodule additional operating modes has limitations with respect to the choice of RED and FED delay with respect to the operating duty cycle of the ePWMxA and ePWMxB outputs.

- The dead-band counters have also been increased to 14 bits
- Deadband and deadband high-resolution registers are now shadowed
- High-resolution deadband RED and FED have been enabled using the DBREDHR and DBFEDHR registers

Note

The PWM chopper is not enabled when high-resolution deadband is enabled.

High-resolution deadband RED and FED requires half-cycle clocking mode (DBCTL[HALFCYCLE] = 1).

Cannot have both RED and FED together applied to both ePWMxA and ePWMxB. RED and FED together can be applied only to either OutA OR OutB.

Phase shifting B-channel with respect to the A-channel: When PWMxB is derived from PWMxA using the DEDB_MODE bit and by delaying rising edge and falling edge by the phase shift amount. When the duty cycle value on PWMxA is less than this phase shift amount, PWMxA's falling edge has precedence over the delayed rising edge for PWMxB. Make sure the duty cycle value of the current waveform applied to the dead-band module is greater than the required phase shift amount.

The Type 4 action qualifier and dead-band outputs of the ePWM module are delayed by one TBCLK cycle in comparison to the Type 2 ePWM module, although the Type 4 behavior is the same as the Type 3 PWM. Both PWMA and PWMB signals are delayed under all circumstances.

Shadow Mode:

The shadow mode for the DBRED is enabled by setting the DBCTL[SHDWDBREDMODE] bit and the shadow register for DBFED is enabled by setting the DBCTL [SHDWDBFEDMODE] bit. Shadow mode is disabled by default for both DBRED and DBFED

If the shadow register is enabled, then the content of the shadow register is transferred to the active register on one of the following events as specified by the DBCTL [LOADREDMODE] and DBCTL [LOADFEDMODE] register bits:

- CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD).
- CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00)
- Both CTR = PRD and CTR = Zero

The DBCTL register can be shadowed. The shadow mode for DBCTL is enabled by setting the DBCTL2[SHDWDBCTLMODE] bit. If the shadow register is enabled then the content of the shadow register is transferred to the active register on one of the following events as specified by the DBCTL2[LOADDBCTLMODE] register bit:

- CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD)
- CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00)
- Both CTR = PRD and CTR = Zero

Note

The application software must enable shadow load mode in the DBCTL[SHDWDBREDMODE] and DBCTL[SHDWDBFEDMODE] **before** programming values for the DBRED and DBFED registers. If the shadow register is enabled **after** programming the DBRED and DBFED registers, the DBRED and DBFED registers are loaded with a value of 0.

Global Load Support

Global load control mechanism can also be used for DBRED:DBREDHR, DBFED:DBFEDHR, and DBCTL registers by configuring the appropriate bits in the global load configuration register (GLDCFG). When global load mode is selected the transfer of contents from shadow register to active register, for all registers that have this mode enabled, occurs at the same event as defined by the configuration bits in the Global Shadow to Active Load Control Register (GLDCTL). The Global load control mechanism is explained in Section 30.4.7.

Note

When DBRED/DBFED active is loaded with a new shadow value while DB counters are counting, the new DBRED/DBFED value only affects the NEXT PWMx edge and not the current edge.

A Deadband value of zero cannot be used when the Global Shadow to Active Load is set to occur at CTR=ZERO. Similarly, a Deadband value of PRD cannot be used when the Global Shadow to Active Load is set to occur at CTR=PRD.

30.8.3 Operational Highlights for the Dead-Band Submodule

The configuration options for the dead-band submodule are shown in Figure 30-41.

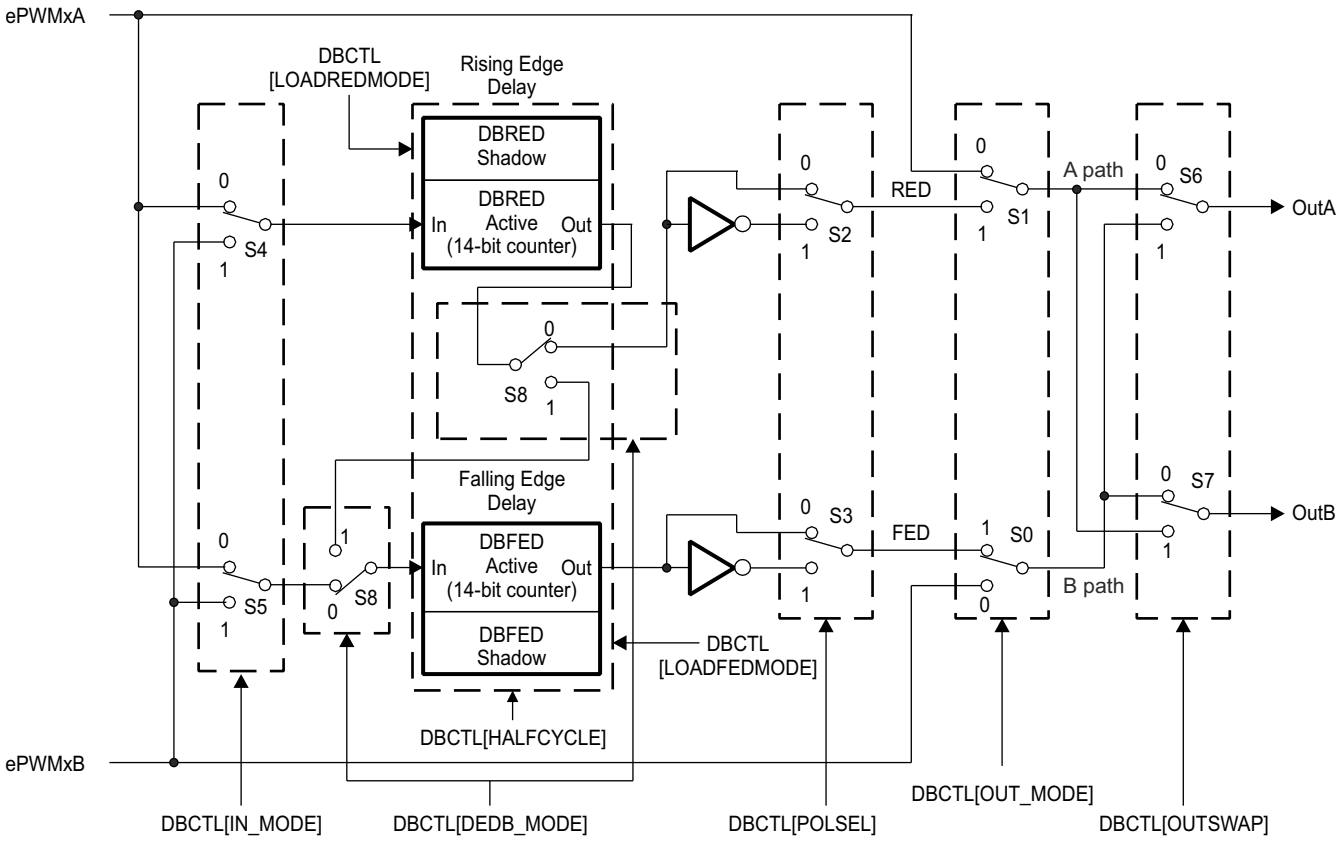


Figure 30-41. Configuration Options for the Dead-Band Submodule

Although all combinations are supported, not all are typical usage modes. Table 30-10 documents some classical dead-band configurations. These modes assume that the DBCTL[IN_MODE] is configured such that EPWMxA In is the source for both falling-edge and rising-edge delay. Enhanced, or non-traditional modes can be achieved by changing the input signal source. The modes shown in Table 30-10 fall into the following categories:

- **Mode 1: Bypass both falling-edge delay (FED) and rising-edge delay (RED):** Allows the user to fully disable the dead-band submodule from the PWM signal path.
- **Mode 2-5: Classical Dead-Band Polarity Settings:** These represent typical polarity configurations that can address all the active-high and active-low modes required by available industry power switch gate drivers. The waveforms for these typical cases are shown in Figure 30-42. Note that to generate equivalent waveforms to Figure 30-42, configure the action-qualifier submodule to generate the signal as shown for EPWMxA.
- **Mode 6: Bypass rising-edge delay (RED) and Mode 7: Bypass falling-edge delay (FED):** Finally the last two entries in Table 30-10 show combinations where either the falling-edge delay (FED) or rising-edge delay (RED) blocks are bypassed.

Figure 30-42 shows waveforms for typical cases where 0% < duty < 100%.

Table 30-10. Classical Dead-Band Operating Modes

Mode	Mode Description	DBCTL[POLSEL]		DBCTL[OUT_MODE]	
		S3	S2	S1	S0
1	EPWMxA and EPWMxB Passed Through (No Delay)	X	X	0	0
2	Active High Complementary (AHC)	1	0	1	1
3	Active Low Complementary (ALC)	0	1	1	1
4	Active High (AH)	0	0	1	1
5	Active Low (AL)	1	1	1	1
6	EPWMxA Out = EPWMxA In (No Delay) EPWMxB Out = EPWMxA In with Falling-Edge Delay	0 or 1	0 or 1	0	1
7	EPWMxA Out = EPWMxA In with Rising-Edge Delay EPWMxB Out = EPWMxB In with No Delay	0 or 1	0 or 1	1	0

Table 30-11. Additional Dead-Band Operating Modes

Mode Description	DBCTL[DEDB-MODE]	DBCTL[OUTSWAP]	
	S8	S6	S7
EPWMxA and EPWMxB signals are as defined by OUT-MODE bits.	0	0	0
EPWMxA = A-path as defined by OUT-MODE bits. EPWMxB = A-path as defined by OUT-MODE bits (rising-edge delay or delay-bypassed A-signal path)	0	0	1
EPWMxA = B-path as defined by OUT-MODE bits (falling-edge delay or delay-bypassed B-signal path) EPWMxB = B-path as defined by OUT-MODE bits	0	1	0
EPWMxA = B-path as defined by OUT-MODE bits (falling-edge delay or delay-bypassed B-signal path) EPWMxB = A-path as defined by OUT-MODE bits (rising-edge delay or delay-bypassed A-signal path)	0	1	1
Rising-edge delay applied to EPWMxA / EPWMxB as selected by S4 switch (IN-MODE bits) on A signal path only. Falling-edge delay applied to EPWMxA / EPWMxB as selected by S5 switch (IN-MODE bits) on B signal path only.	0	X	X
Rising-edge delay and falling-edge delay applied to source selected by S4 switch (IN-MODE bits) and output to B signal path only.(1)	1	X	X

(1) When this bit is set to 1, the user can always either set OUT_MODE bits such that Apath = InA or set OUTSWAP bits such that EPWMxA=Bpath. Otherwise, EPWMxA is invalid.

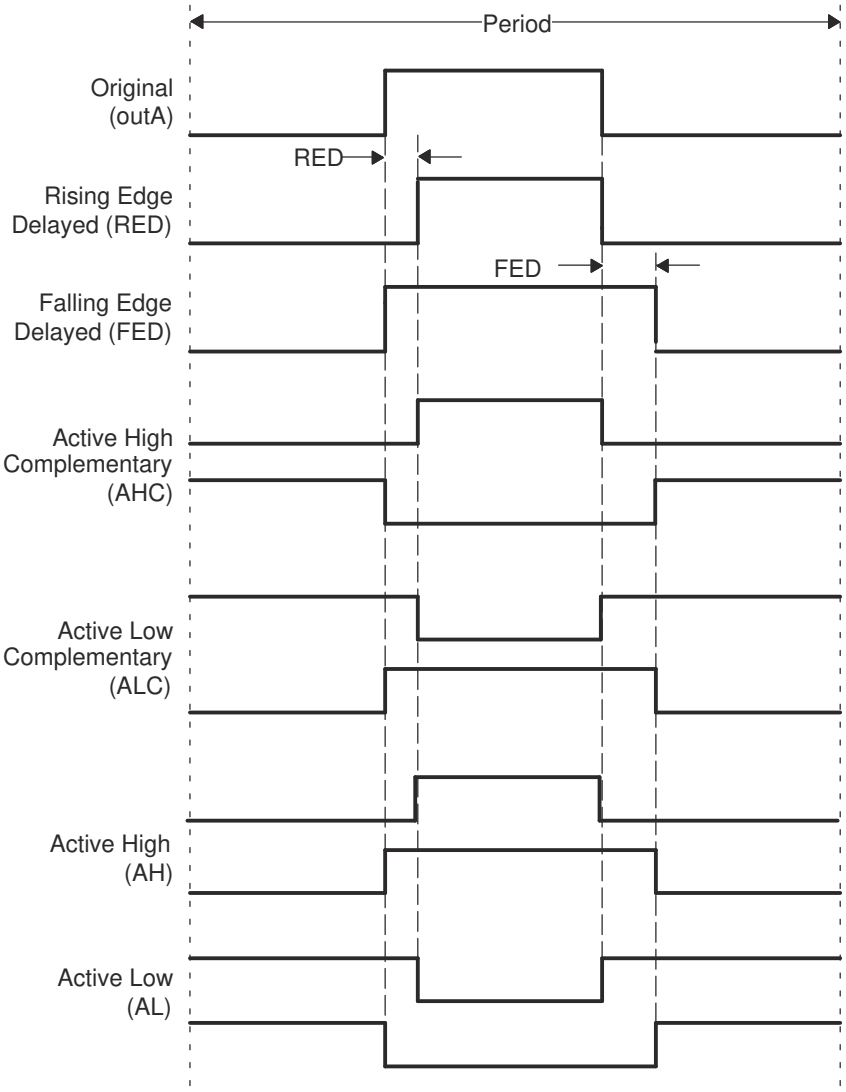


Figure 30-42. Dead-Band Waveforms for Typical Cases (0% < Duty < 100%)

The dead-band submodule supports independent values for rising-edge (RED) and falling-edge (FED) delays. The amount of delay is programmed using the DBRED and DBFED registers. These are 10-bit registers and the value represents the number of TBCLK (time-base clock) pulses by which a signal edge is delayed. For example, the formula to calculate falling-edge-delay and rising-edge-delay is:

$FED = DBFED \times T_{TBCLK}$

$RED = DBRED \times T_{TBCLK}$

Where T_{TBCLK} is the period of TBCLK, the prescaled version of EPWMCLK.

For convenience, delay values for various TBCLK options are shown in Table 30-12. The ePWM input clock frequency that these delay values been computed by is 100MHz.

Table 30-12. Dead-Band Delay Values in µs as a Function of DBFED and DBRED

Dead-Band Value	Dead-Band Delay (µs)		
DBFED, DBRED	TBCLK = EPWMCLK/1	TBCLK = EPWMCLK /2	TBCLK = EPWMCLK/4
1	0.01	0.02	0.04
5	0.05	0.10	0.20
10	0.10	0.20	0.40
100	1.00	2.00	4.00
200	2.00	4.00	8.00
400	4.00	8.00	16.00
500	5.00	10.00	20.00
600	6.00	12.00	24.00
700	7.00	14.00	28.00
800	8.00	16.00	32.00
900	9.00	18.00	36.00
1000	10.00	20.00	40.00

When half-cycle clocking is enabled, the formula to calculate the falling-edge-delay and rising-edge-delay becomes:

FED = DBFED × $T_{TBCLK}/2$

RED = DBRED × $T_{TBCLK}/2$

30.9 PWM Chopper (PC) Submodule

The PWM chopper submodule allows a high-frequency carrier signal to modulate the PWM waveform generated by the action-qualifier and dead-band submodules. This capability is important if pulse transformer-based gate drivers to control the power switching elements are needed.

Figure 30-43 illustrates the PWM chopper submodule within the ePWM.

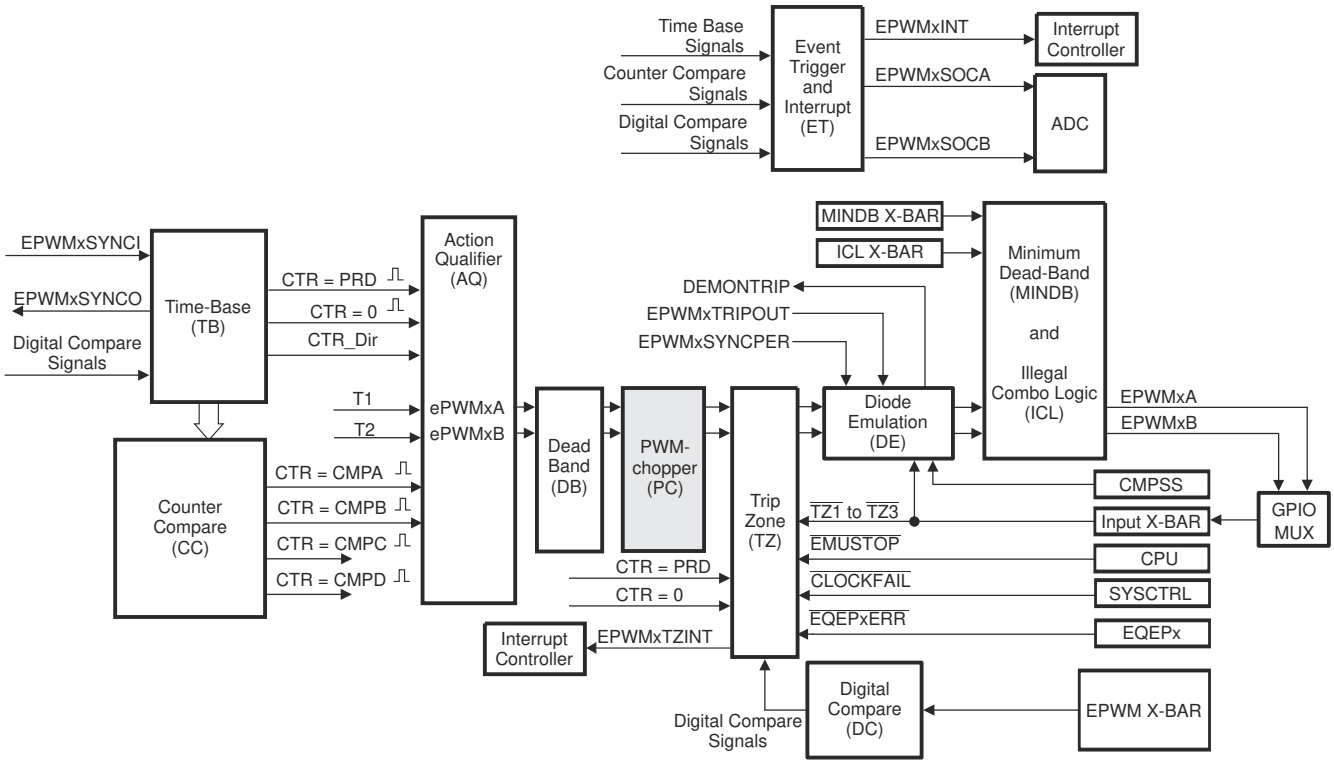


Figure 30-43. PWM Chopper Submodule

30.9.1 Purpose of the PWM Chopper Submodule

The key functions of the PWM chopper submodule are:

- Programmable chopping (carrier) frequency
- Programmable pulse width of first pulse
- Programmable duty cycle of second and subsequent pulses
- Can be fully bypassed if not required

30.9.2 Operational Highlights for the PWM Chopper Submodule

Figure 30-44 shows the operational details of the PWM chopper submodule. The carrier clock is derived from EPWMCLK. The clock frequency and duty cycle are controlled using the CHPFREQ and CHPDUTY bits in the PCCTL register. The one-shot block is a feature that provides a high energy first pulse to make sure hard and fast power switch turn on, while the subsequent pulses sustain pulses, making sure the power switch remains on. The one-shot width is programmed using the OSHTWTH bits. The PWM chopper submodule can be fully disabled (bypassed) using the CHPEN bit.

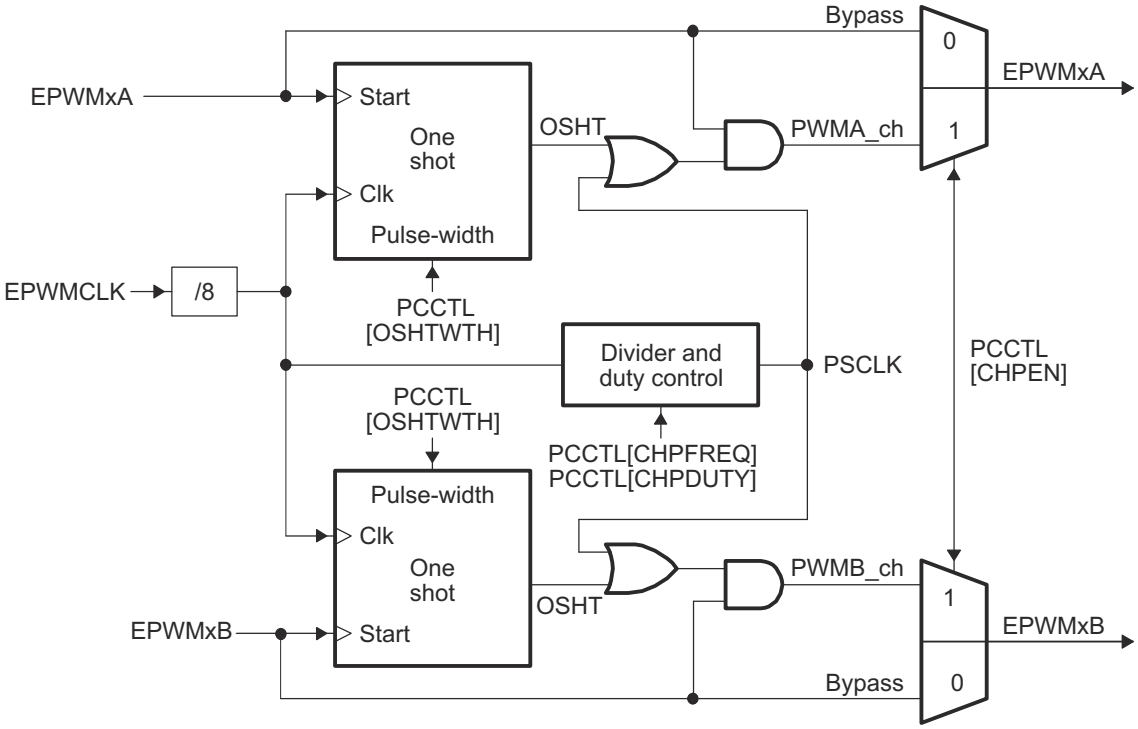


Figure 30-44. PWM Chopper Submodule Operational Details

30.9.3 Waveforms

Figure 30-45 shows simplified waveforms of the chopping action only; one-shot and duty-cycle control are not shown. Details of the one-shot and duty-cycle control are discussed in the following sections.

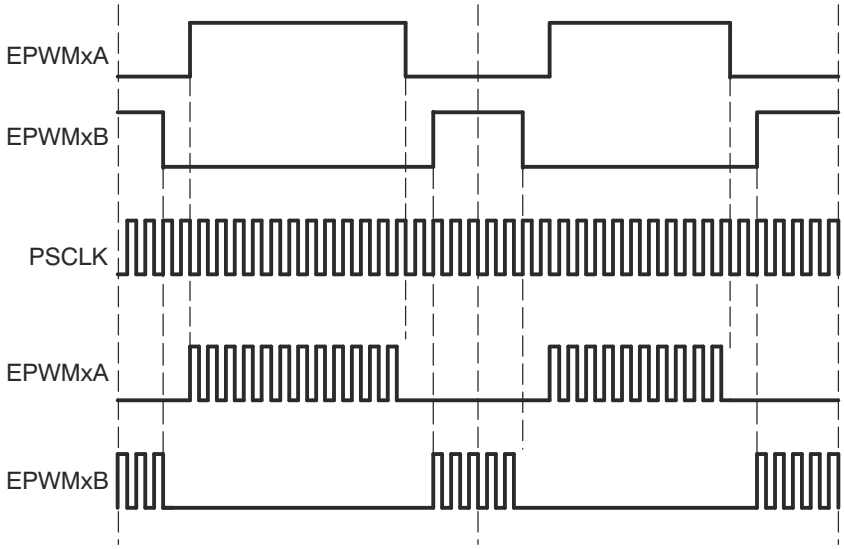


Figure 30-45. Simple PWM Chopper Submodule Waveforms Showing Chopping Action Only

30.9.3.1 One-Shot Pulse

The width of the first pulse can be programmed to any of 16 possible pulse width values. The width or period of the first pulse is given by:

$$T_{1stpulse} = T_{EPWMCLK} \times 8 \times OSHTWTH$$

Where $T_{EPWMCLK}$ is the period of the system clock (EPWMCLK) and OSHTWTH is the four control bits (value from 1 to 16)

Figure 30-46 shows the first and subsequent sustaining pulses and Table 30-13 gives the possible pulse width values for a EPWMCLK = 80MHz.

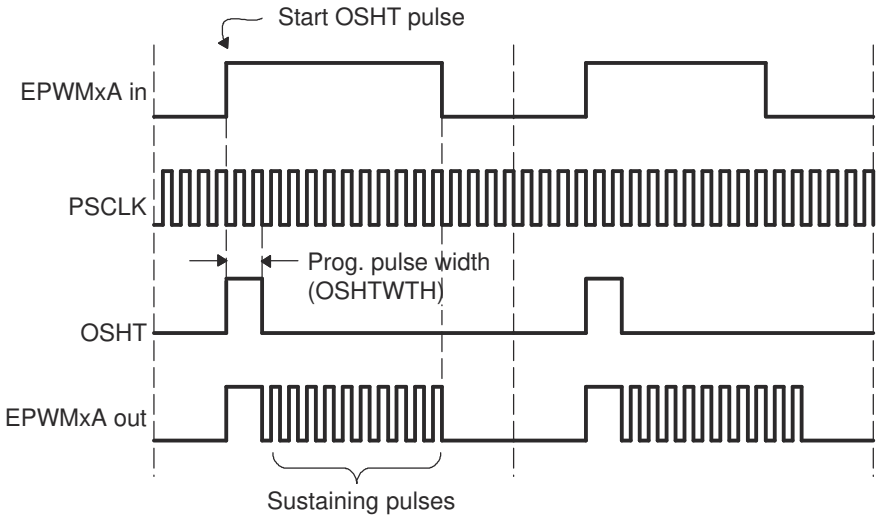


Figure 30-46. PWM Chopper Submodule Waveforms Showing the First Pulse and Subsequent Sustaining Pulses

Table 30-13. Possible Pulse Width Values for EPWMCLK = 80MHz

OSHTWTH (hex)	Pulse Width (ns)
0	100
1	200
2	300
3	400
4	500
5	600
6	700
7	800
8	900
9	1000
A	1100
B	1200
C	1300
D	1400
E	1500
F	1600

30.9.3.2 Duty Cycle Control

Pulse transformer-based gate drive designs need to comprehend the magnetic properties or characteristics of the transformer and associated circuitry. Saturation is one such consideration. To assist the gate drive designer, the duty cycles of the second and subsequent pulses have been made programmable. These sustaining pulses make sure the correct drive strength and polarity is maintained on the power switch gate during the on period, and hence a programmable duty cycle allows a design to be tuned or optimized using software control.

Figure 30-47 shows the duty cycle control that is possible by programming the CHPDUTY bits. One of seven possible duty ratios can be selected ranging from 12.5% to 87.5%.

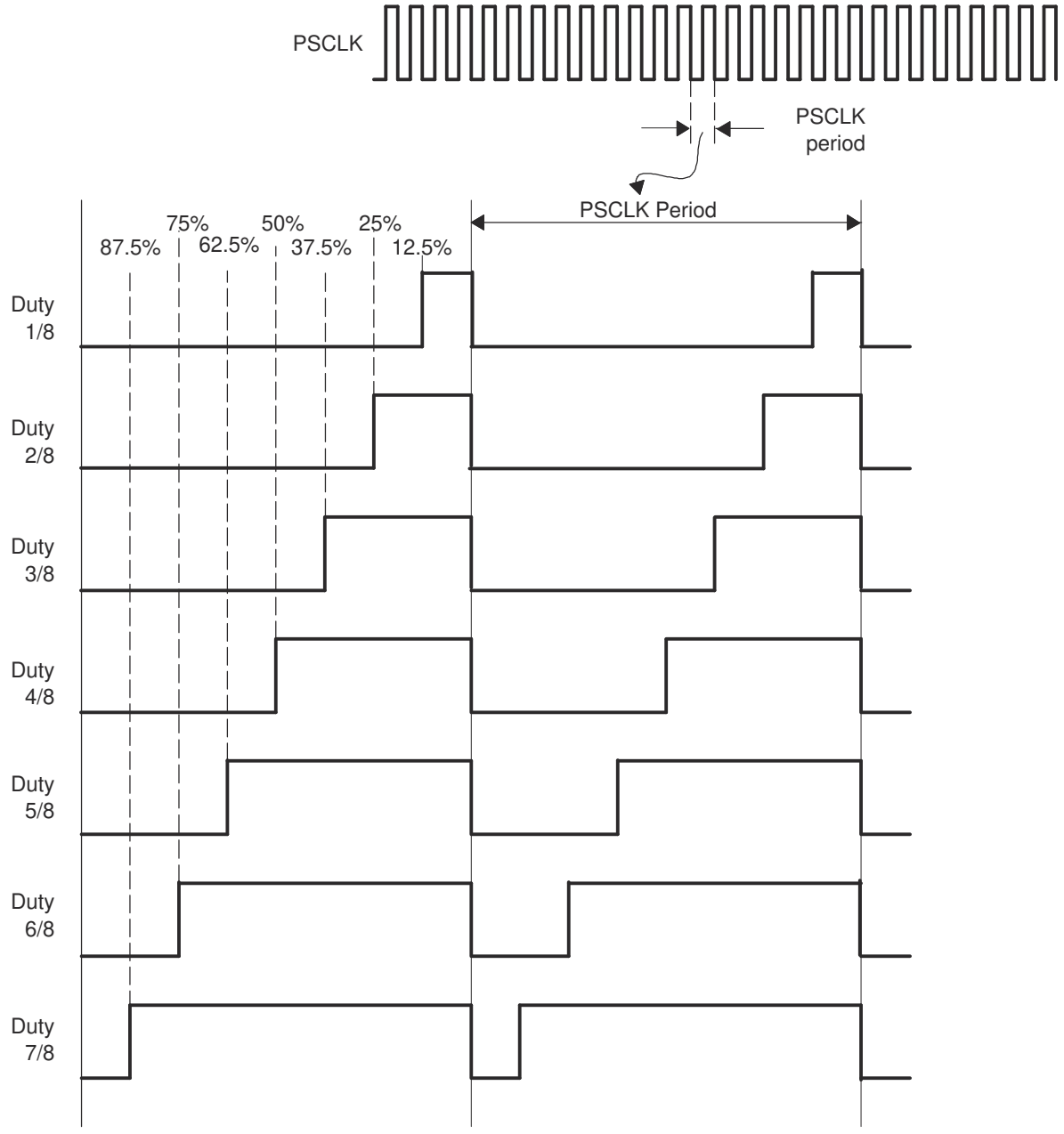


Figure 30-47. PWM Chopper Submodule Waveforms Showing the Pulse Width (Duty Cycle) Control of Sustaining Pulses

30.10 Trip-Zone (TZ) Submodule

Each ePWM module is connected to six $\overline{TZn}$ signals ($\overline{TZ1}$ to $\overline{TZ6}$). $\overline{TZ1}$ to $\overline{TZ3}$ are sourced from the GPIO mux. $\overline{TZ4}$ is sourced from an inverted EQEPxERR signal on those devices with an EQEP module. $\overline{TZ5}$ is connected to the system clock fail logic, and $\overline{TZ6}$ is sourced from the EMUSTOP output from the CPU. These signals indicate external fault or trip conditions, and the ePWM outputs can be programmed to respond accordingly when faults occur.

Figure 30-48 illustrates the trip-zone submodule within the ePWM.

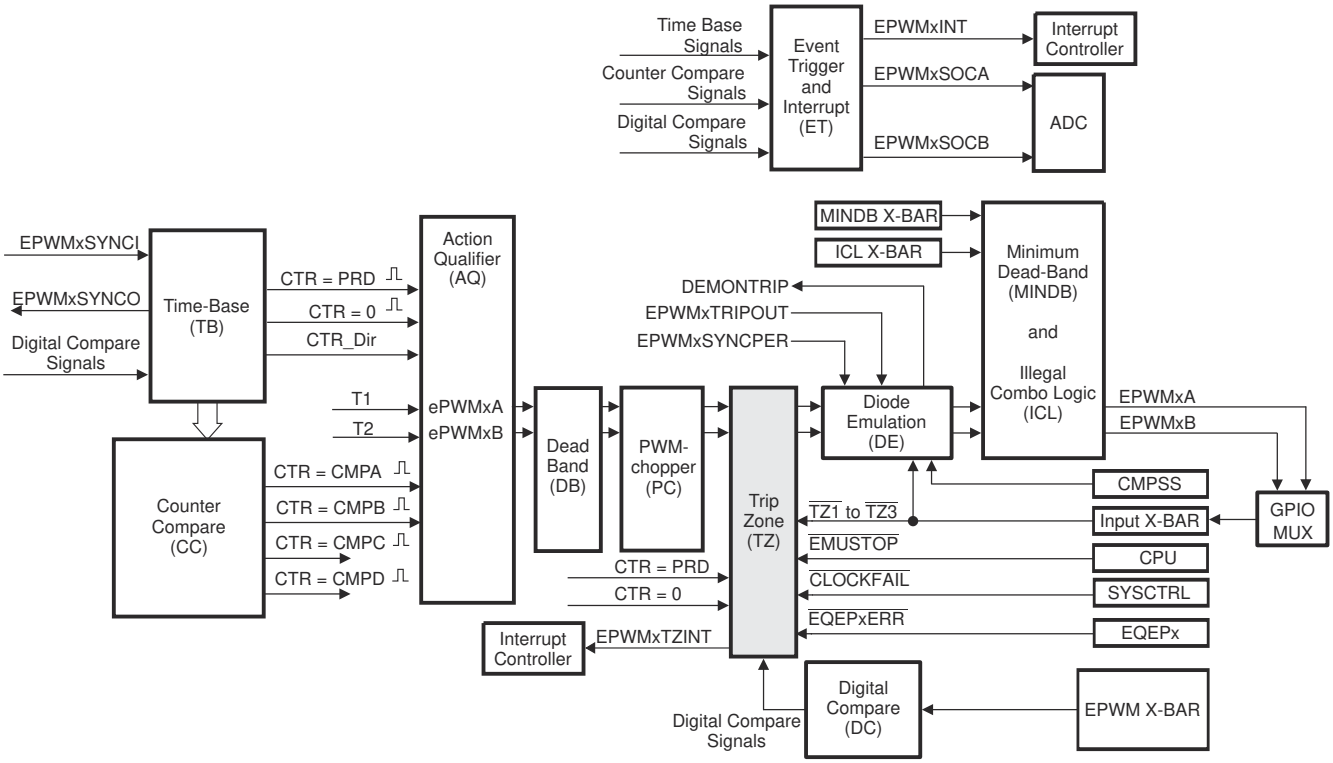


Figure 30-48. Trip-Zone Submodule

30.10.1 Purpose of the Trip-Zone Submodule

The key functions of the trip-zone submodule are:

- Trip inputs $\overline{TZ1}$ to $\overline{TZ6}$ can be flexibly mapped to any ePWM module.
- Upon a fault condition, outputs EPWMxA and EPWMxB can be forced to one of the following:
  - High
  - Low
  - High-impedance
  - No action taken
- Support for one-shot trip (OSHT) for major short circuits or over-current conditions.
- Support for cycle-by-cycle tripping (CBC) for current limiting operation.
- Support for digital compare tripping (DC) based on state of on-chip analog comparator module outputs and $\overline{TZ1}$ to $\overline{TZ3}$ signals.
- Each trip-zone input and digital compare (DC) submodule DCAEVT1/2 or DCBEVT1/2 force event can be allocated to either one-shot or cycle-by-cycle operation.
- Interrupt generation is possible on any trip-zone input.
- Software-forced tripping is also supported.
- The trip-zone submodule can be fully bypassed if the trip-zone submodule is not required.

30.10.2 Operational Highlights for the Trip-Zone Submodule

The following sections describe the operational highlights and configuration options for the trip-zone submodule.

The trip-zone signals $\overline{TZ1}$ to $\overline{TZ6}$ (also collectively referred to as $\overline{TZn}$) are active-low input signals. When one of these signals goes low, or when a DCAEVT1/2 or DCBEVT1/2 force happens based on the TZDCSEL register event selection, the indication is that a trip event has occurred. Each ePWM module can be individually configured to ignore or use each of the trip-zone signals or DC events. Which trip-zone signals or DC events are used by a particular ePWM module is determined by the TZSEL register for that specific ePWM module. The trip-zone signals can or cannot be synchronized to the ePWMclock (EPWMCLK) and digitally filtered within the GPIO MUX block. A minimum of 3*TBCLK low pulse width on $\overline{TZn}$ inputs is sufficient to trigger a fault condition on the ePWM module. If the pulse width is less than this, the trip condition cannot be latched by CBC or OST latches. The asynchronous trip makes sure that if clocks are missing for any reason, the outputs can still be tripped by a valid event present on $\overline{TZn}$ inputs. The GPIOs or peripherals must be appropriately configured. For more information, see the *System Control and Interrupts* chapter.

Each $\overline{TZn}$ input can be individually configured to provide either a cycle-by-cycle or one-shot trip event for an ePWM module. DCAEVT1 and DCBEVT1 events can be configured to directly trip an ePWM module or provide a one-shot trip event to the module. Likewise, DCAEVT2 and DCBEVT2 events can also be configured to directly trip an ePWM module or provide a cycle-by-cycle trip event to the module. This configuration is determined by the TZSEL[DCAEVT1/2], TZSEL[DCBEVT1/2], TZSEL[CBCn], and TZSEL[OSHTn] control bits (where n corresponds to the trip input), respectively.

- **Cycle-by-Cycle (CBC):**

  When a cycle-by-cycle trip event occurs, the action specified in the TZCTL[TZA] and TZCTL[TZB] bits is carried out immediately on the EPWMxA and EPWMxB outputs. Table 30-14 lists some of the possible actions. Independent actions can be specified based on the occurrence of the event while the counter is counting up or while the counter is counting down by appropriately configuring bits in the TZCTL2 register. Actions specified in the TZCTL2 register take effect only when the ETZE bit in TZCTL2 is set.

  Additionally, when a cycle-by-cycle trip event occurs, the cycle-by-cycle trip event flag (TZFLG[CBC]) is set and a EPWMx_TZINT interrupt is generated when enabled in the TZEINT register and interrupt controller. A corresponding flag for the event that caused the CBC event is also set in register TZCBCFLG.

  If the CBC interrupt is enabled using the TZEINT register and DCAEVT2 or DCBEVT2 are selected as CBC trip sources using the TZSEL register, it is not necessary to also enable the DCAEVT2 or DCBEVT2 interrupts in the TZEINT register, as the DC events trigger interrupts through the CBC mechanism.

  The specified condition on the inputs is automatically cleared based on the selection made with TZCLR[CBCPULSE] if the trip event is no longer present. Therefore, in this mode, the trip event is cleared or reset every PWM cycle. The TZFLG[CBC] and TZCBCFLG flag bits remain set until the flag bits are manually cleared by writing to the TZCLR[CBC] and TZCBCCLR flag bits. If the cycle-by-cycle trip event is still present when the TZFLG[CBC] and TZCBCFLG register bits are cleared, then these bits are again immediately set.

- **One-Shot (OSHT):**

  When a one-shot trip event occurs, the action specified in the TZCTL[TZA] and TZCTL[TZB] bits is carried out immediately on the EPWMxA and EPWMxB output. Table 30-14 lists some of the possible actions. Independent actions can be specified based on the occurrence of the event while the counter is counting up and while the counter is counting down by appropriately configuring bits in TZCTL2 register. Actions specified in TZCTL2 register take effect only when ETZE bit in TZCTL2 is set.

  Additionally, when a one-shot trip event occurs, the one-shot trip event flag (TZFLG[OST]) is set and a EPWMx_TZINT interrupt is generated when enabled in the TZEINT register and interrupt controller. A corresponding flag for the event that caused the OST event is also set in register TZOSTFLG. The one-shot trip condition must be cleared manually by writing to the TZCLR[OST] bit. If desired, the TZOSTFLG register bit can be cleared by manually writing to the corresponding bit in the TZOSTCLR register.

If the one-shot interrupt is enabled using the TZEINT register and DCAEVT1 or DCBEVT1 are selected as OSHT trip sources using the TZSEL register, it is not necessary to also enable the DCAEVT1 or DCBEVT1 interrupts in the TZEINT register, as the DC events trigger interrupts through the OSHT mechanism.

Note

Clear the TZFLG and TZOSTFLG flags after making sure that the TRIPIN source of the OST has become inactive. Otherwise, if interrupts are enabled, depending on when the flags are cleared, an OST interrupt can occur and the OST flags are zero.

- **Digital Compare Events (DCAEVT1/2 and DCBEVT1/2):**

  A digital compare DCAEVT1/2 or DCBEVT1/2 event is generated based on a combination of the DCAH/DCAL and DCBH/DCBL signals as selected by the TZDCSEL register. The signals which source the DCAH/DCAL and DCBH/DCBL signals are selected using the DCTRIPSEL register and can be either trip zone input pins or analog comparator CMPSSx signals. For more information on the digital compare submodule signals, see Section 30.14.

  When a digital compare event occurs, the action specified in the TZCTL[DCAEVT1/2] and TZCTL[DCBEVT1/2] bits is carried out immediately on the EPWMxA and EPWMxB output. Table 30-14 lists the possible actions. Independent actions can be specified based on the occurrence of the event while the counter is counting up and while the counter is counting down by appropriately configuring bits in TZCTLDCA and TZCTLDCB register. Actions specified in TZCTLDCA and TZCTLDCB registers take effect only when ETZE bit in TZCTL2 is set.

  In addition, the relevant DC trip event flag (TZFLG[DCAEVT1/2] / TZFLG[DCBEVT1/2]) is set and a EPWMx_TZINT interrupt is generated when enabled in the TZEINT register and interrupt controller.

  The specified condition on the pins is automatically cleared when the DC trip event is no longer present. The TZFLG[DCAEVT1/2] or TZFLG[DCBEVT1/2] flag bit remains set until the flag is manually cleared by writing to the TZCLR[DCAEVT1/2] or TZCLR[DCBEVT1/2] bit. If the DC trip event is still present when the TZFLG[DCAEVT1/2] or TZFLG[DCBEVT1/2] flag is cleared, then the flag is again immediately set.

- **Edge detection within a programmable TBCTR range (CAPEVT):**

  An edge detection within a programmable TBCTR range is added in type 5 ePWM. When a CAPIN edge does not occur within a specified range of TBCTR values, the CAPEVT signal is generated. The TBCTR range during which a CAPIN edge must occur is determined by XMINMAX_ACTIVE register. A gating signal CAPGATE can also be used to gate the CAPIN edge. For more information on the CAPEVT signal, see Section 30.14.4.4.

  In addition, the EPWMx_TZINT interrupt is generated when enabled in the TZEINT register and interrupt controller.

  The TZFLG[CAPEVT] flag bit remains set until the flag is manually cleared by writing to the TZCLR[CAPEVT] bit. If the CAPEVT event is still present when the TZFLG[CAPEVT] flag is cleared, then the flag is again immediately set.

The action taken when a trip event occurs can be configured individually for each of the ePWM output pins by way of the TZCTL, TZCTL2, TZCTLDCA, and TZCTLDCB register bit fields. Some of the possible actions, shown in Table 30-14, can be taken on a trip event.

The trip signal generated by the ePWM module can be selected through the TZTRIPOUTSEL register. This register has an ORed version of all the enabled trip signals. The TRIPOUT signal is routed to eCAP Trip Mux,PWM-XBAR and Output-XBAR.

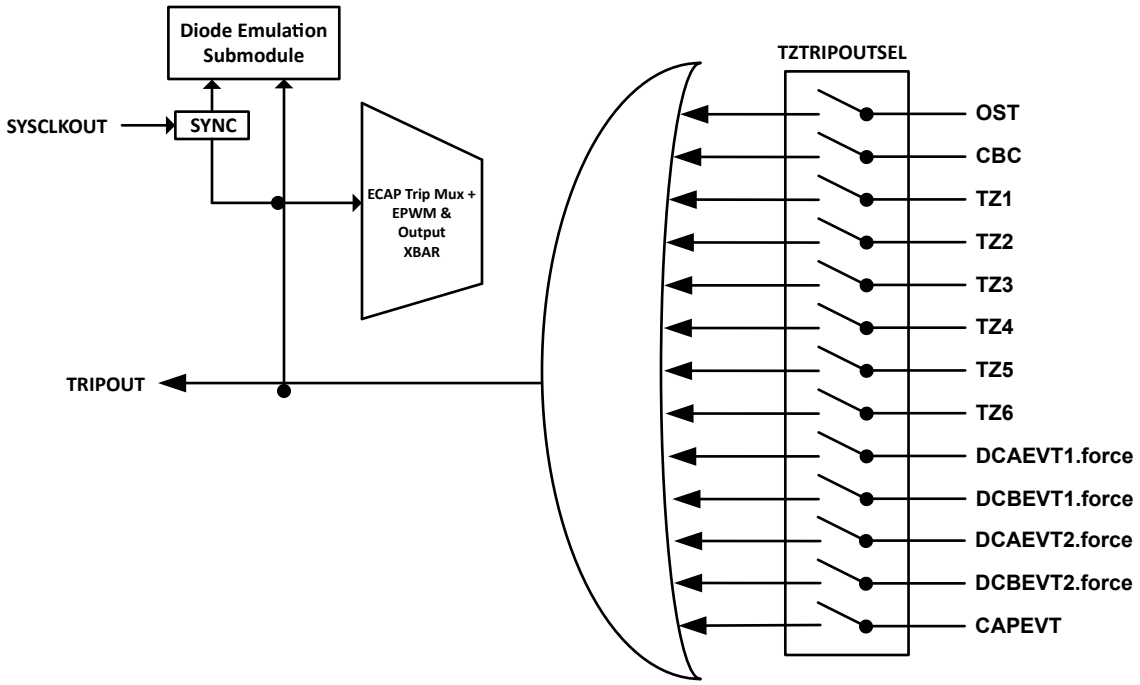


Figure 30-49. Trip-Zone TRIPOUT Selection

Table 30-14. Possible Actions On a Trip Event

TZCTL Register Bitfield Settings	EPWMxA and EPWMxB	Comment
0,0	High-Impedance	Tripped
0,1	Force to High State	Tripped
1,0	Force to Low State	Tripped
1,1	No Change	Do Nothing. No change is made to the output.

Example 30-1. Trip-Zone Configurations

Scenario A:
A one-shot trip event on $\overline{TZ1}$ pulls both EPWM1A, EPWM1B low and also forces EPWM2A and EPWM2B high.

- Configure the ePWM1 registers as follows:
  - TZSEL[OSHT1] = 1: enables $\overline{TZ1}$ as a one-shot event source for ePWM1
  - TZCTL[TZA] = 2: EPWM1A is forced low on a trip event.
  - TZCTL[TZB] = 2: EPWM1B is forced low on a trip event.
- Configure the ePWM2 registers as follows:
  - TZSEL[OSHT1] = 1: enables $\overline{TZ1}$ as a one-shot event source for ePWM2
  - TZCTL[TZA] = 1: EPWM2A is forced high on a trip event.
  - TZCTL[TZB] = 1: EPWM2B is forced high on a trip event.

Scenario B:
A cycle-by-cycle event on $\overline{TZ5}$ pulls both EPWM1A, EPWM1B low.
A one-shot event on $\overline{TZ1}$ or $\overline{TZ6}$ puts EPWM2A into a high impedance state.

- Configure the ePWM1 registers as follows:
  - TZSEL[CBC5] = 1: enables $\overline{TZ5}$ as a cycle-by-cycle event source for ePWM1
  - TZCTL[TZA] = 2: EPWM1A is forced low on a trip event.
  - TZCTL[TZB] = 2: EPWM1B is forced low on a trip event.
- Configure the ePWM2 registers as follows:
  - TZSEL[OSHT1] = 1: enables $\overline{TZ1}$ as a one-shot event source for ePWM2
  - TZSEL[OSHT6] = 1: enables $\overline{TZ6}$ as a one-shot event source for ePWM2
  - TZCTL[TZA] = 0: EPWM2A is put into a high-impedance state on a trip event.
  - TZCTL[TZB] = 3: EPWM2B ignores the trip event.

Note

When configuring the GPIOs and INPUT X-BAR/EPWM X-BAR options, be aware that a change in the X-BAR input selections can cause an unwanted event. Therefore, set up the GPIO and X-BAR input configurations before enabling the ePWM Trip-Zone. If a requirement is to change the GPIO/X-BAR configurations while the ePWM Trip-Zone is enabled, the user can turn off the TRIPs by clearing the TZSEL register and reconfiguring the TRIP selection (TZSEL) after the INPUT XBAR selection is changed.

30.10.3 Generating Trip Event Interrupts

Figure 30-50 and Figure 30-51 illustrate the trip-zone submodule control and interrupt logic, respectively. DCAEVT1/2 and DCBEVT1/2 signals are described in further detail in Section 30.14.

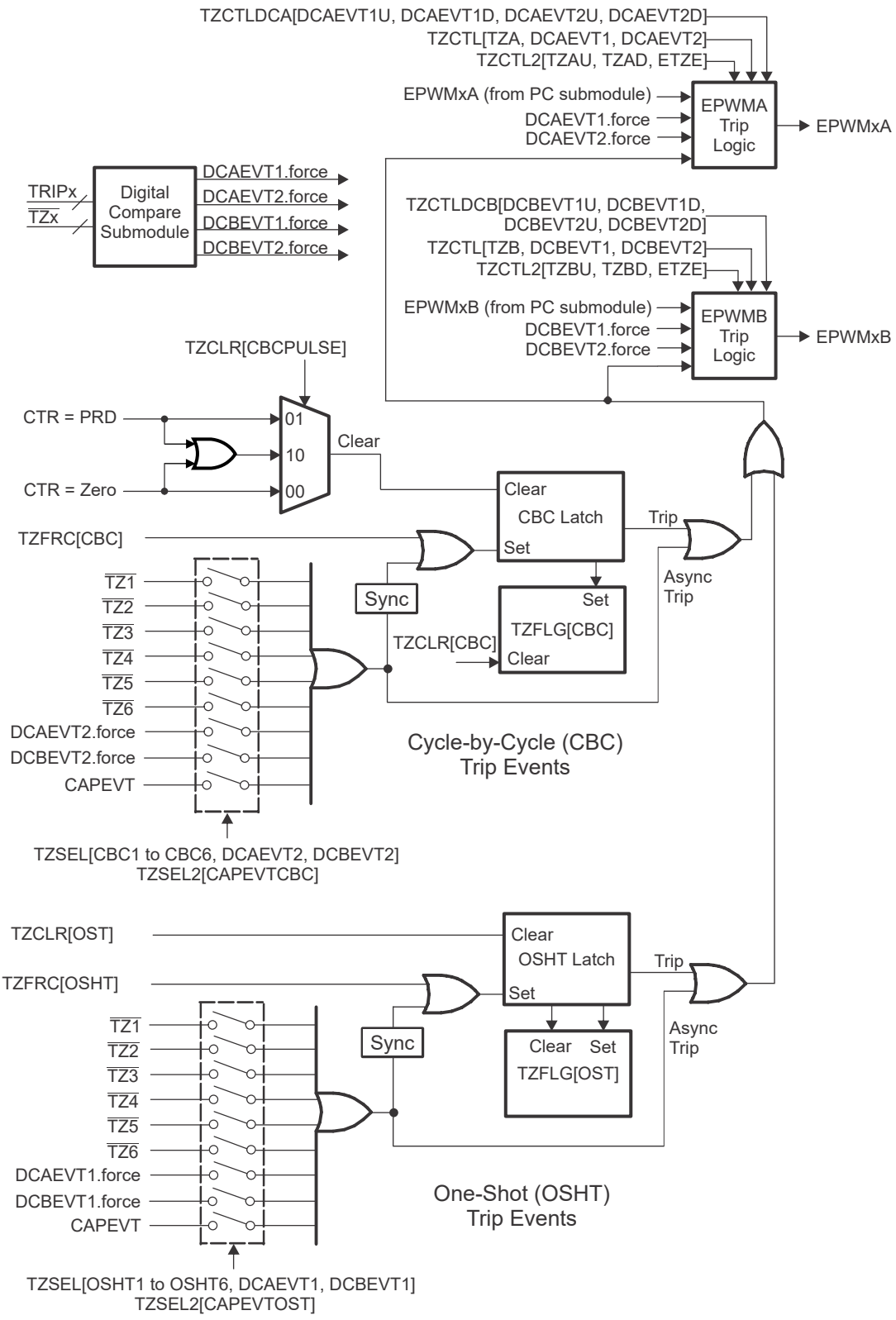


Figure 30-50. Trip-Zone Submodule Mode Control Logic

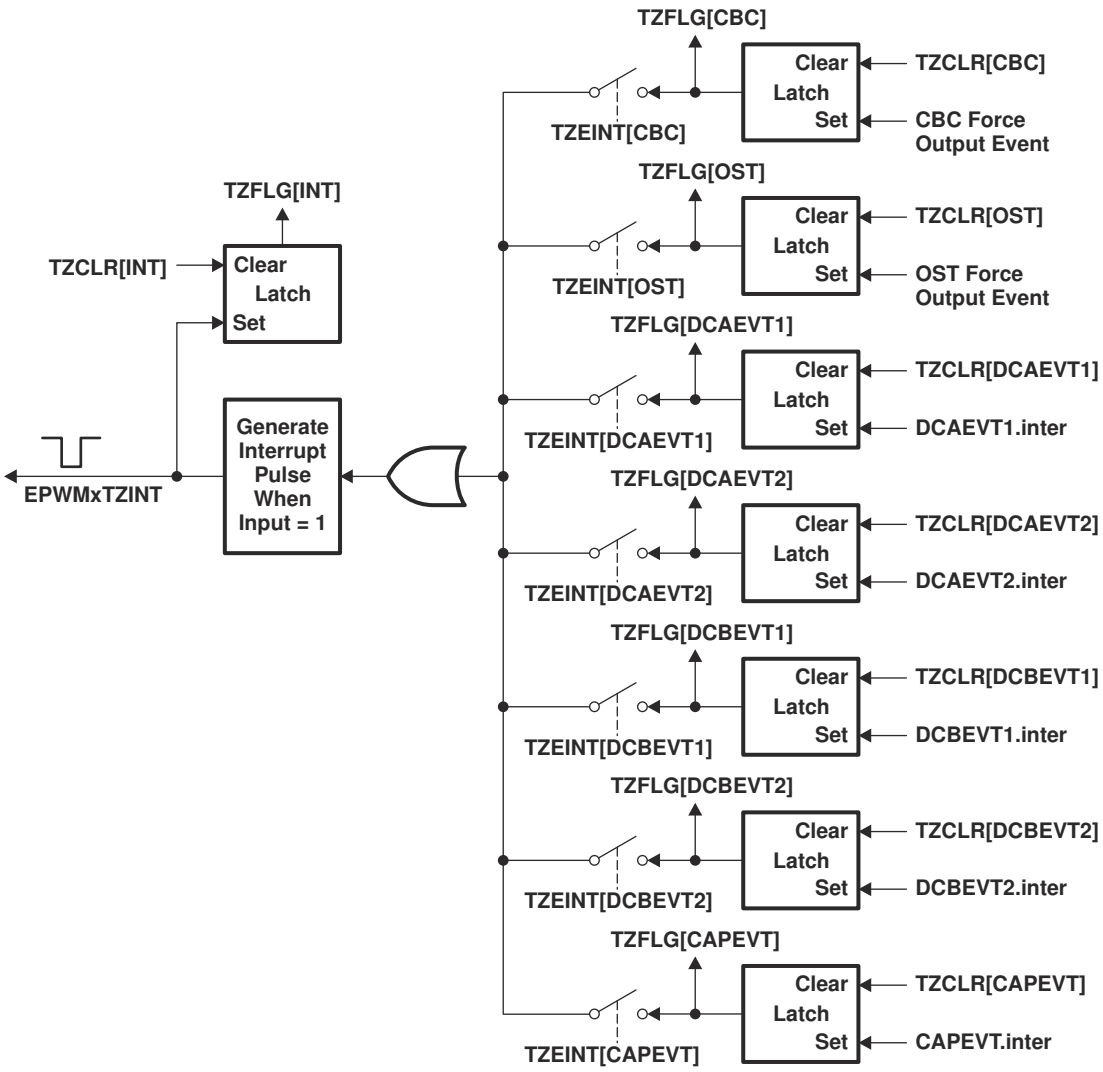


Figure 30-51. Trip-Zone Submodule Interrupt Logic

The signal CAPEVT is generated from the Capture Control Logic and is available in type 5 EPWM.

These individual flags for the CBC, OST and DCxEVTy can be used to detect the source of the EPWMxTZINT Interrupt. When multiple sources are used to generate the EPWMxTZINT interrupt, reading and clearing the flags takes different actions based on the specific event.

30.11 Diode Emulation (DE) Submodule

The purpose of the Diode Emulation logic is to provide hardware features and the necessary hooks into other IPs to implement robust diode mode sense and control in a noisy environment.

Diode Emulation features include:

- Ability to choose any of the comparator outputs as trips to detect entry into DE mode.
- Ability to switch the comparator thresholds, dynamically in hardware upon DE mode entry.
- Two modes of clearing/de-evaluating the DE condition:
  - Software clear
  - Cycle-by-cycle clear on PWMSYNC event
- Configurable source selects of ePWM in DE mode.
- Ability to monitor the DE mode duration and generate a trip event to ePWMs.

Figure 30-52 illustrates the diode emulation submodule within the ePWM.

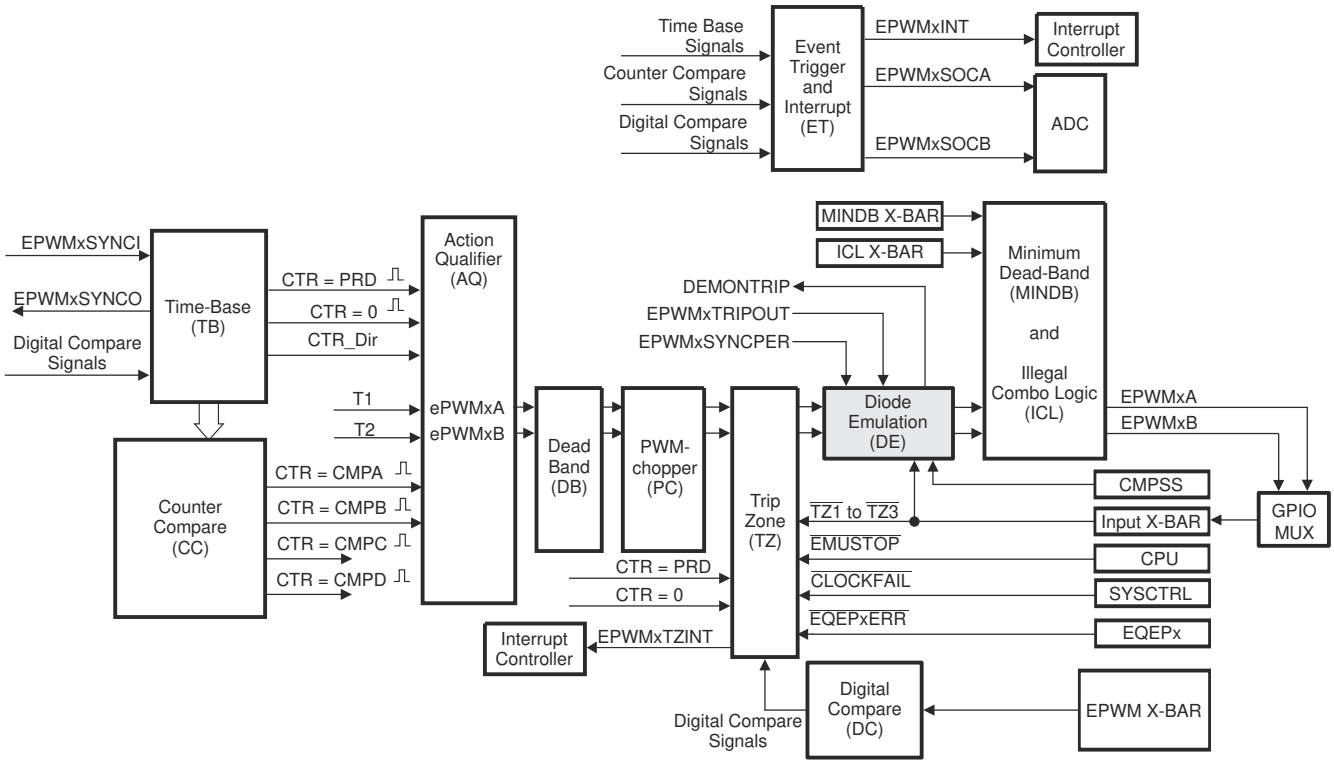


Figure 30-52. Diode Emulation Submodule

Figure 30-53 shows the interfaces to the DE block. As can be seen from the diagram, DE function is associated with an instance of ePWMx. The EPWMxA and EPWMxB signals from a given instance of ePWM module pass through the associated DE block and minimum dead band logic. In addition to EPWMxA and EPWMxB, two signals, EPWMxA_DB_NO_HR and EPWMxB_DB_NO_HR are tapped from the ePWM modules. These two signals are PWM signals that are tapped before the signals pass through the high-resolution delay lines and come from the dead-band submodule outputs. If high-resolution is not used, then EPWMxA_DB_NO_HR and EPWMxB_DB_NO_HR are just the outputs of the dead-band submodule.

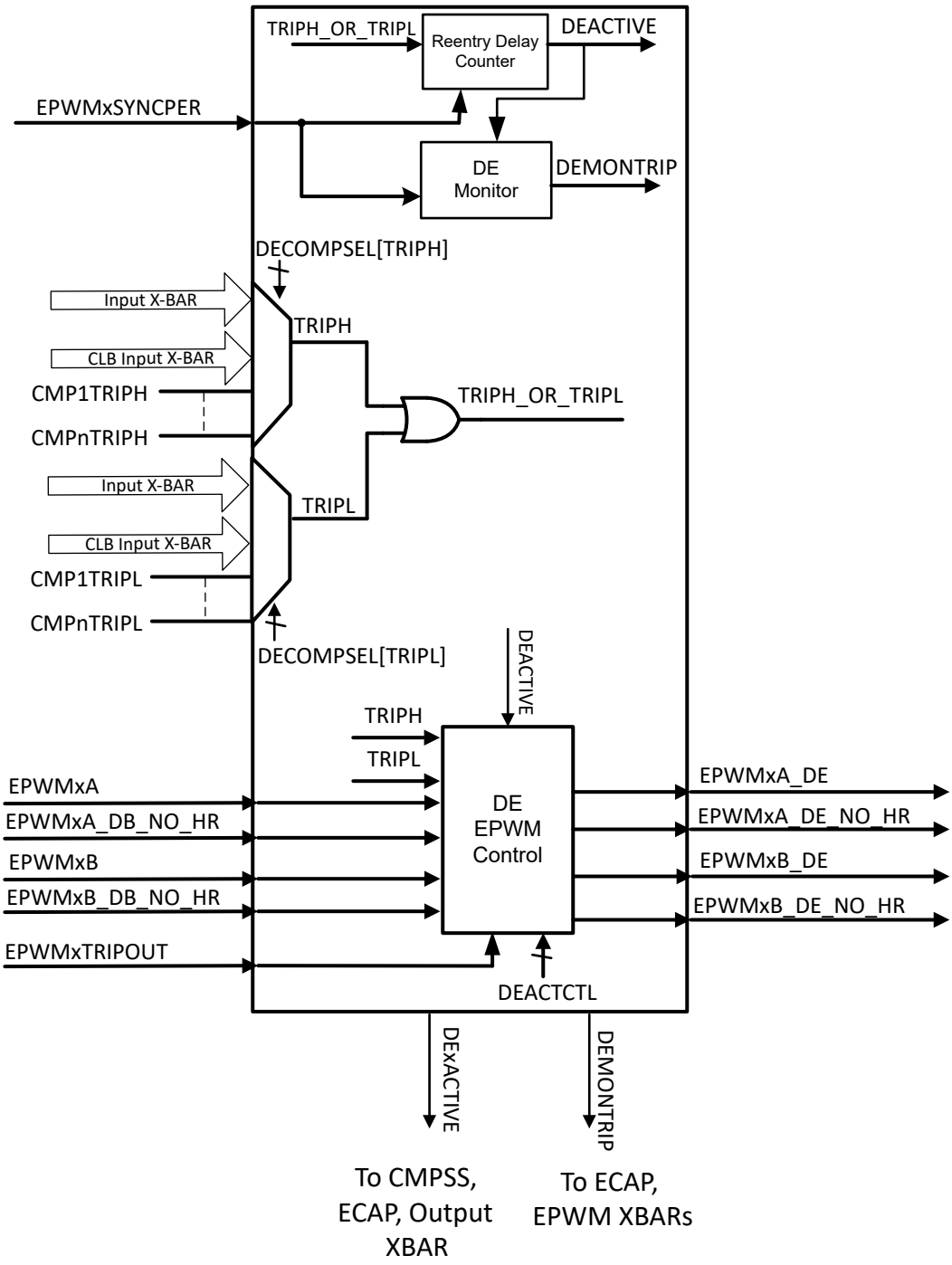


Figure 30-53. Diode Emulation Block Diagram

The DE block can be configured to select one of the comparators or one of the outputs of the Input X-Bar or CLB Input X-Bar, as source of trip signals (TRIPH, TRIPL). The selected comparator is responsible for monitoring the current in the external power converter. Comparator thresholds (High and Low threshold) can be set such that, any breach of these thresholds indicates a need to switch to the DE mode.

Once DE mode is entered, indicated by setting of DEACTIVE flag, the ePWMs sent out of DE block are controlled by configuration registers in the DE block, and are not be the same as ePWMA/B from the associated ePWM instance. Once the DEACTIVE flag is set, the threshold settings of the selected CMPSS are switched to a new set of values (a narrower region). DEACTIVE flag from all of the ePWM instances are hooked up to all the comparator sub-systems (CMPSSy) to enable threshold value switch on DE mode entry. Refer to the CMPSS chapter for more details on how the new threshold values are set.

30.11.1 DEACTIVE Mode

DE mode is entered when TRIPH_OR_TRIPL signal from the selected comparator (CMPSS) goes high. Once the diode emulation mode is entered, typically TRIPH or TRIPL are set and cleared in a sequence (at a given instance of time, TRIPH is high or TRIPL is high – never both) until the current settles within the threshold band.

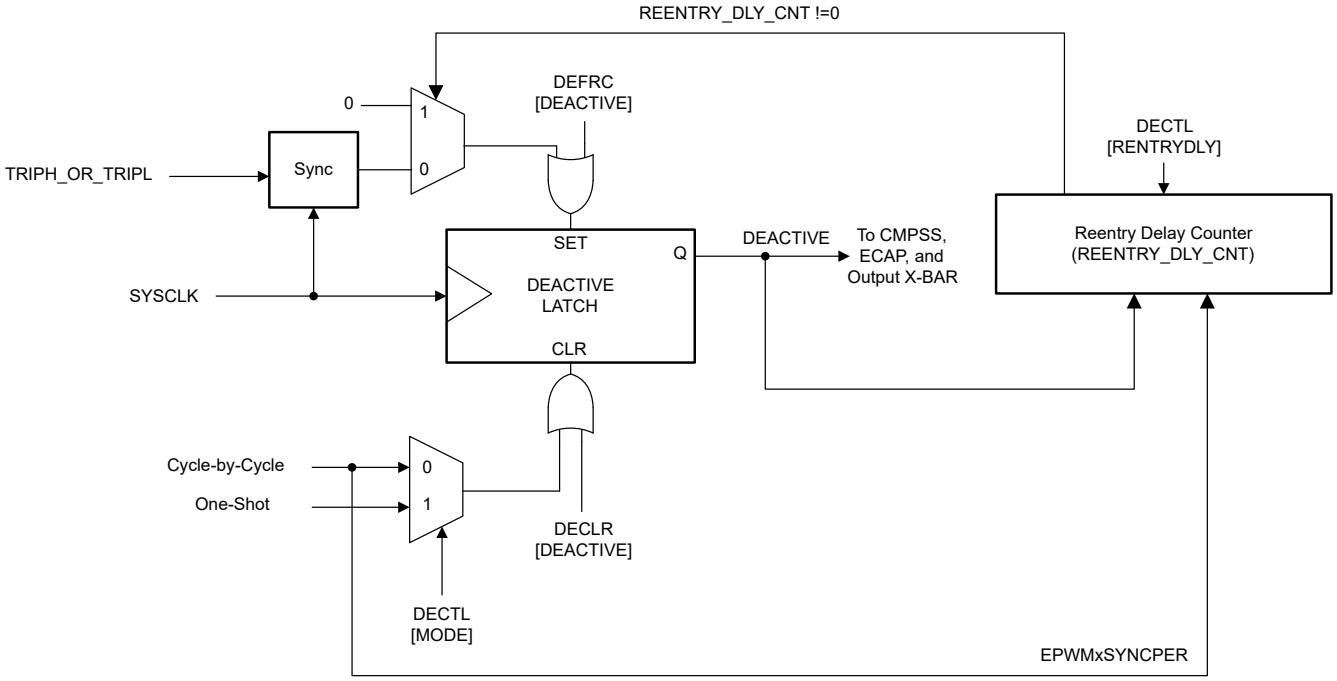


Figure 30-54. DEACTIVE Flag Functionality

Once the DEACTIVE flag is set, the thresholds on CMPSS are changed to an alternate set of thresholds, and also ePWMA/B out of DE function are being controlled by the DEACTCTL register settings. Figure 30-55 demonstrates an example timing diagram illustrating entry into DE mode.

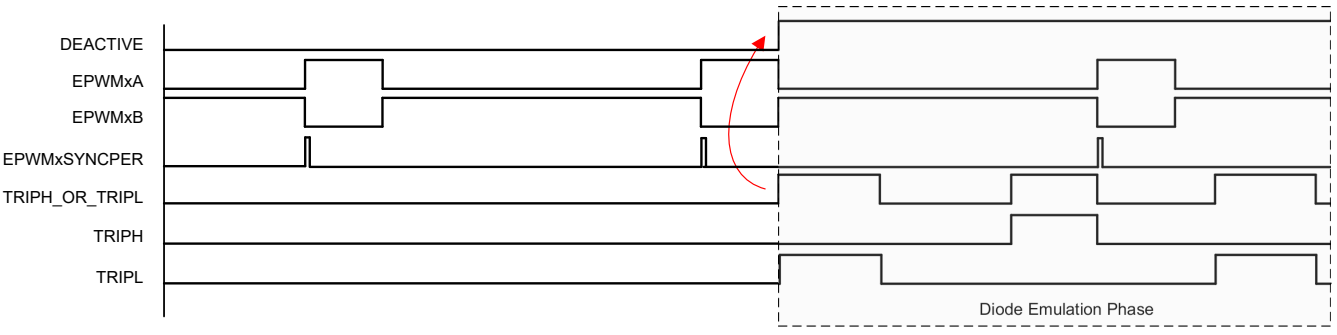


Figure 30-55. Example Timing Sequence Illustrating DE Mode Entry

30.11.2 Exiting DE Mode

DE mode can be exited in two ways, based on the DECTL[MODE] setting:

- Software clear of DEACTIVE flag, DECLR[CLR]
- Cycle-by-cycle clear mode, in which TRIPH_OR_TRIPL is evaluated on every EPWMxSYNCPER and if the trip condition is not present, then DEACTIVE flag is cleared. Figure 30-56 illustrates the clearing of the DEACTIVE flag based on EPWMxSYNCPER.

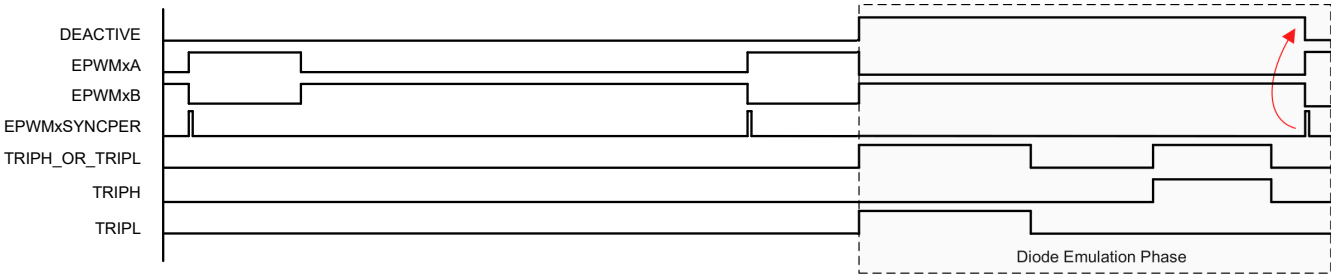


Figure 30-56. Cycle-by-Cycle Mode

30.11.3 Re-Entering DE Mode

Once DE mode is exited, DE mode can be delayed for a certain duration until reentry. This is accomplished by configuring the DECTL[REENTRYDLY] field. REENTRYDLY determines the window in which TRIP signals are prevented from setting the DEACTIVE flag. On a falling edge of DEACTIVE, an internal counter is loaded with the DECTL[REENTRYDLY] value. The counter is decremented on every EPWMxSYNCPER as long as the count value is greater than 0. While the count value is greater than 0, TRIP signals are blocked and DEACTIVE flag is not set even if TRIP events are active.

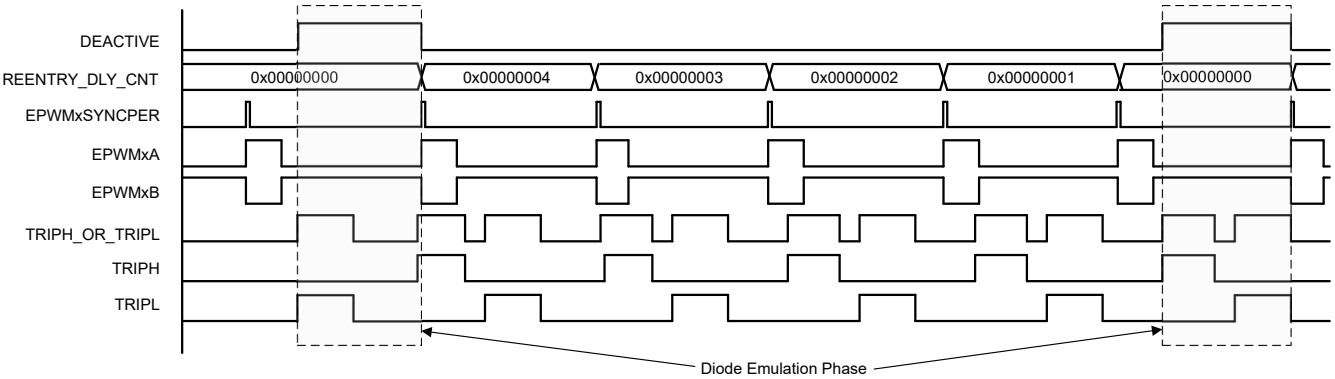


Figure 30-57. DE Mode Reentry Sequence

Figure 30-58 illustrates the circuit driving the EPWMxA/B signals from the DE block. As can be observed, when DEACTIVE flag is not set, EPWMxA_DE, EPWMxB_DE, EPWMxA_DE_NO_HR, and EPWMxB_DE_NO_HR are driven by EPWMA/B and EPWMA/B_DB_NO_HR respectfully. When DEACTIVE flag is set, EPWMA/B_DE are be driven by TRIPH, TRIPL, constant 0, or a constant 1 signal based on the configuration of the DEATCTL[PWMA], DEATCTL[PWMB], DEATCTL[TRIPSELA], DEATCTL[TRIPSELB] fields. When a PWMTRIP signal from the associated ePWM trips, EPWMA/B_DE are be driven by the input PWM signals configured through the DECTL[TRIPENABLE] field.

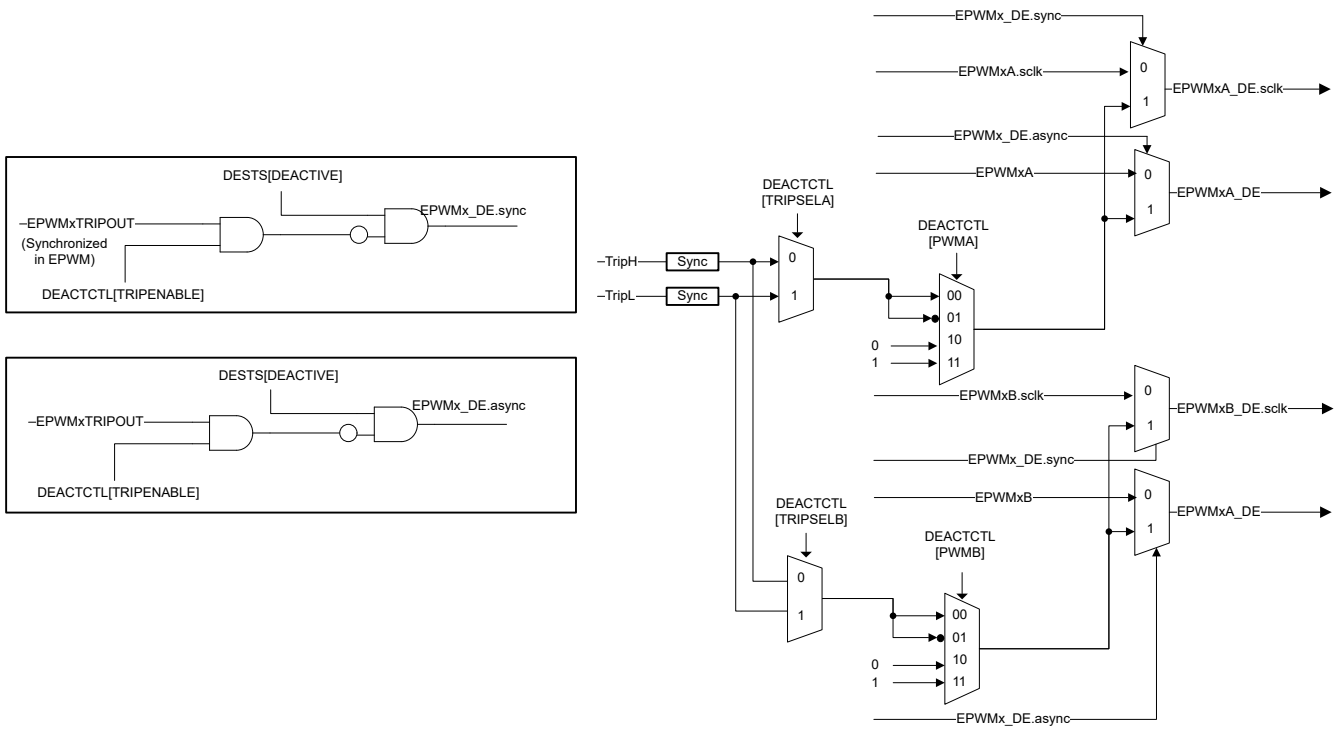


Figure 30-58. Diode Emulation Circuit

Figure 30-59 shows an example waveform, in which DEACTCTL[PWMA] is configured to select TripL as the source, DEACTCTL[PWMB] is configured to select TripH as the source and DEACTCTL[PWMAPOL] and DEACTCTL[PWMBPOL] are both 0.

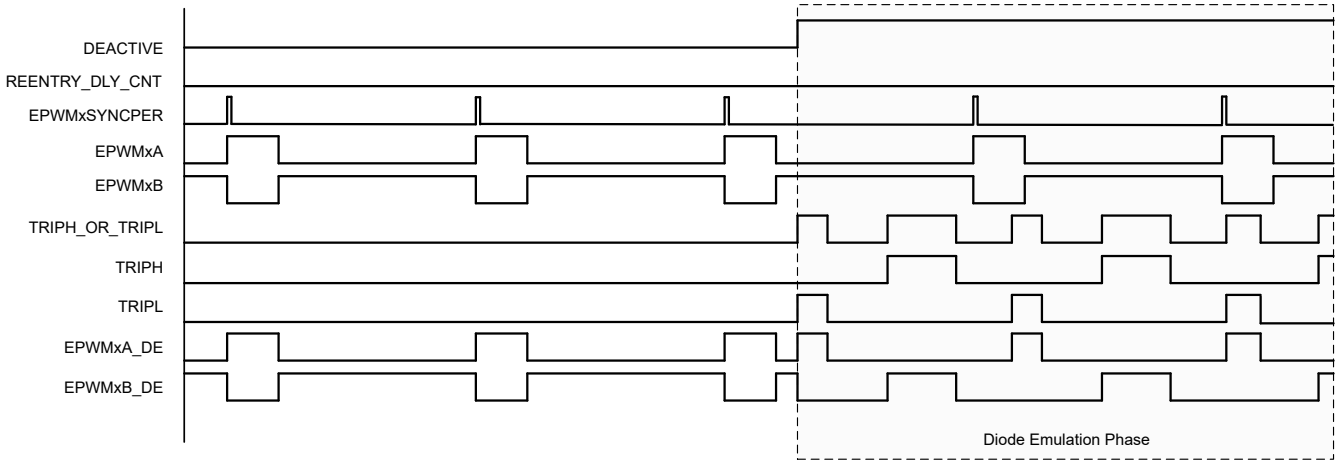


Figure 30-59. Diode Emulation Mode Timing Diagram

30.11.4 DE Monitor

To detect extended DE phase, which is beyond the expected duration, a DE mode monitor counter, DEMONCNT, is provided. This 16-bit counter monitors the frequency of diode mode trip events. The counter if enabled, DEMONCTL[ENABLE], increments on a PWMSYNC event, in steps of DEMONSTEP[INCSTEP] when TRIPH_OR_TRIPL is high, and decrements on a PWMSYNC event, in steps of DEMONSTEP[DECSTEP] when TRIPH_OR_TRIPL is low. If counter exceeds DEMONTHRES[THRESHOLD], then a DEMONTRIP pulse is generated and the counter is cleared. The counter value is saturated to 0 during an underflow and 0xffff on an overflow. The counter is cleared when DECTL[ENABLE] is cleared.

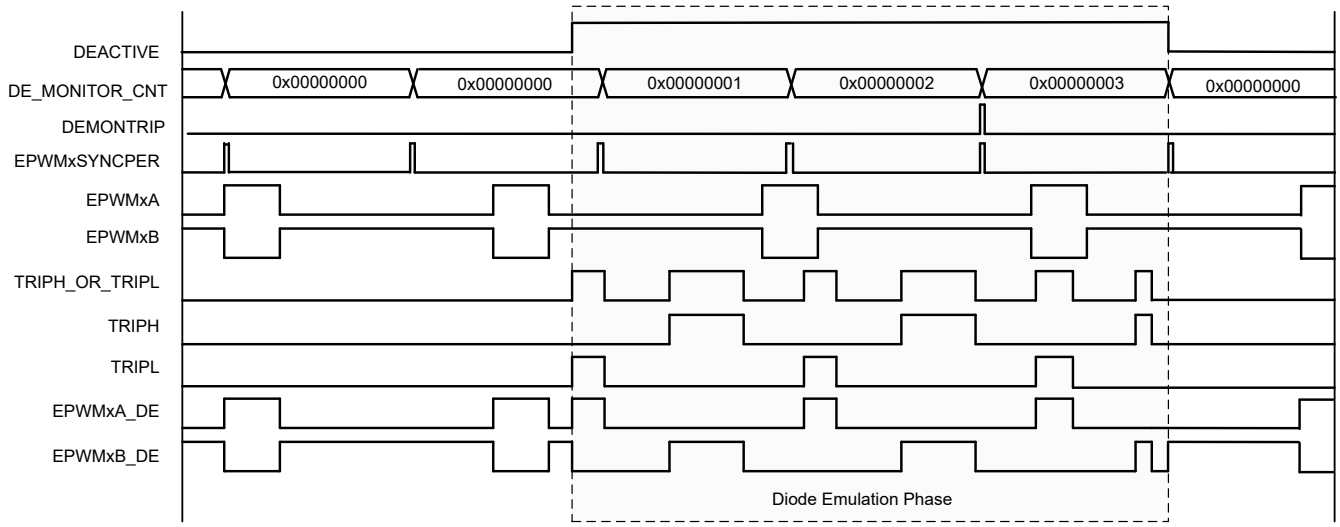


Figure 30-60. DE Mode Monitor Sequence

30.12 Minimum Dead-Band (MINDB) + Illegal Combination Logic (ICL) Submodules

Figure 30-61 illustrates the minimum dead-band and illegal combo submodule within the ePWM.

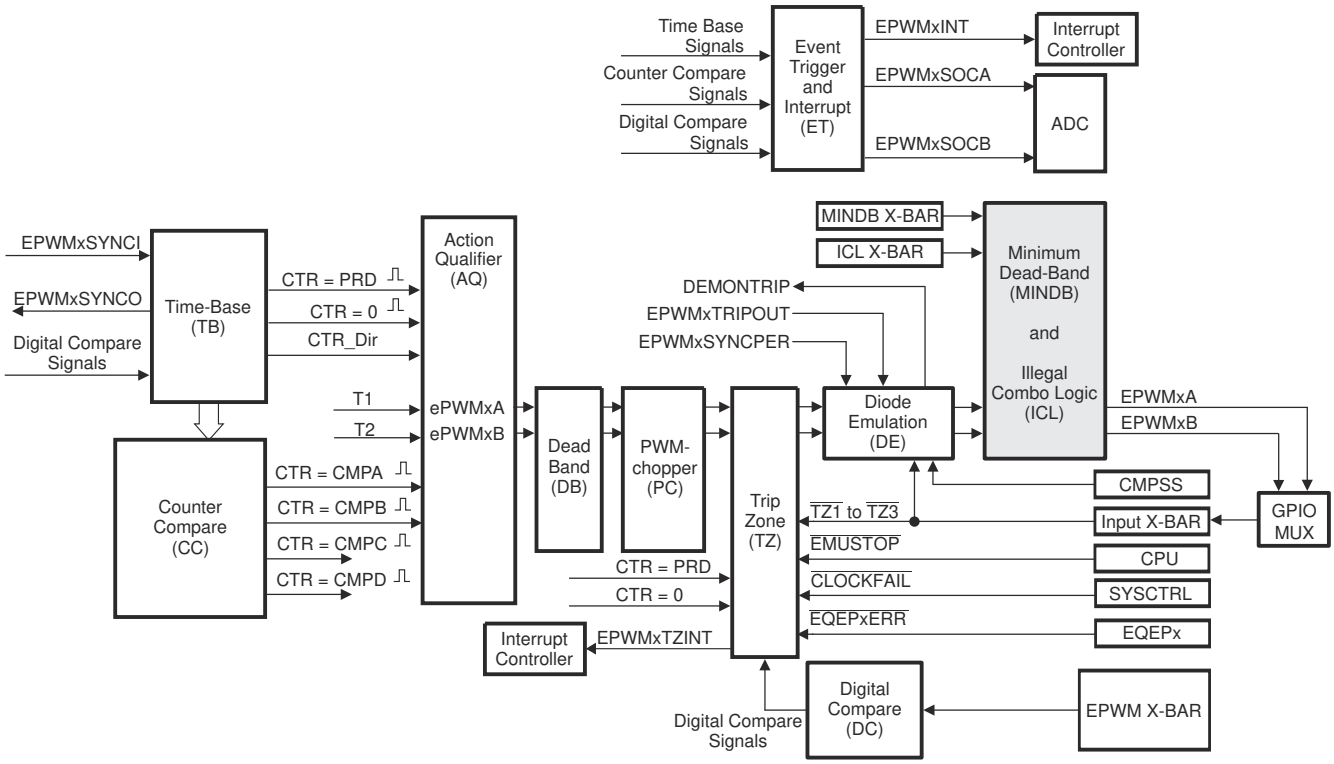


Figure 30-61. Minimum Dead-Band and Illegal Combo Logic Submodule

30.12.1 Minimum Dead-Band (MINDB)

To make sure that the minimum dead band property is not violated, as the application switches between normal mode and DE mode and due to the PWMs potentially switching based on trip inputs, a minimum dead band circuitry show in Figure 30-62 is required.

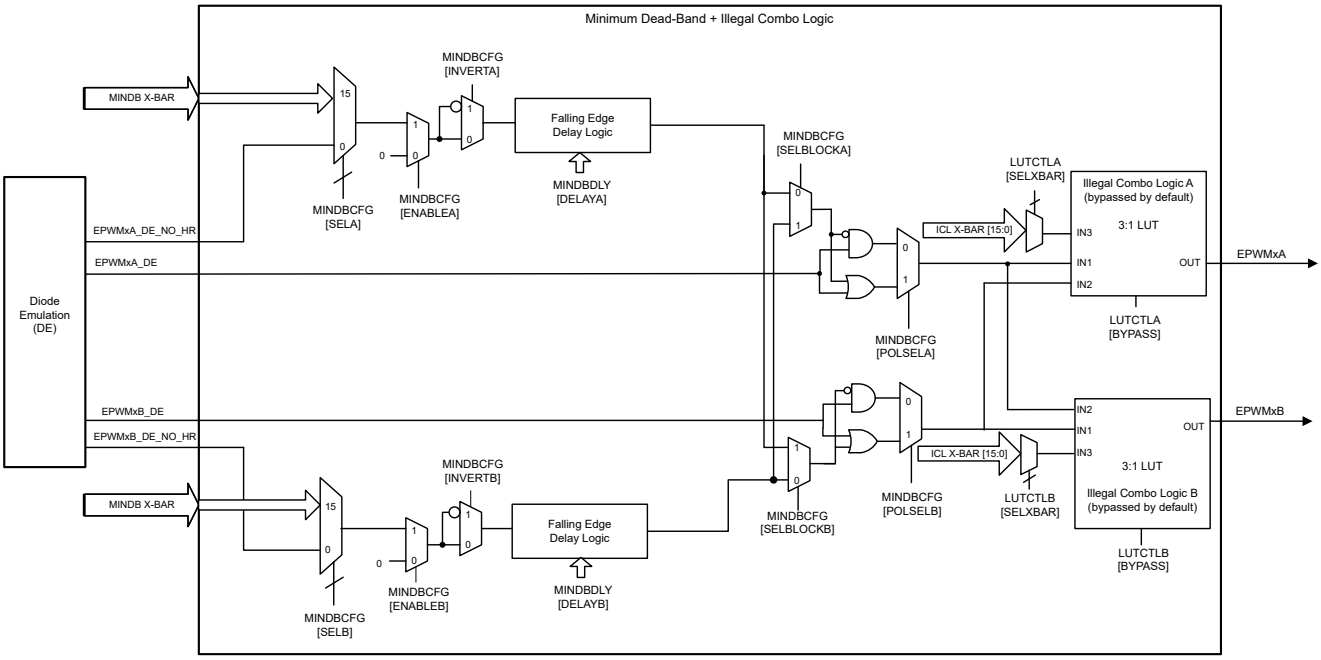


Figure 30-62. Minimum Dead-Band and Illegal Combo Logic Block Diagram

The minimum dead band block provides the ability to configure the minimum dead band duration between a complimentary set of PWMs.

Minimum dead-band logic involves generating a blocking signal (BLOCKA, BLOCKB) after the falling edge of the EPWMA/B_DE. These block signals are used to block transition on the other signal. The input to BLOCKA(B) signal generators is configurable. Normally the sources are EPWMA/B_DB_NO_HR. However, there is a provision provided to select any of the MINDB X-BAR outputs. This provides flexibility to support some of the other application scenarios.

The selected source is fed to the BLOCK signal generation logic. Block signal generation involves, detecting the falling edge based on which BLOCK signal goes high and stretching the BLOCK signal for DELAYA/B cycles, which are software configurable.

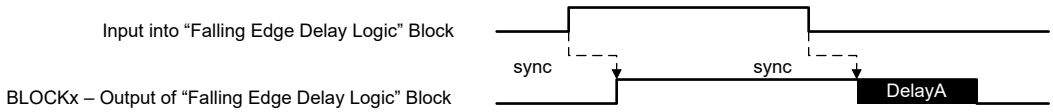


Figure 30-63. Minimum Dead-Band Block Signal Generation

In Figure 30-64 and Figure 30-65, EPWMxA_DE and BLOCKB are getting ANDed and EPWMxB_DE and BLOCKA are getting ANDed.

Note

Red shade is indicative of incorrect scenario.

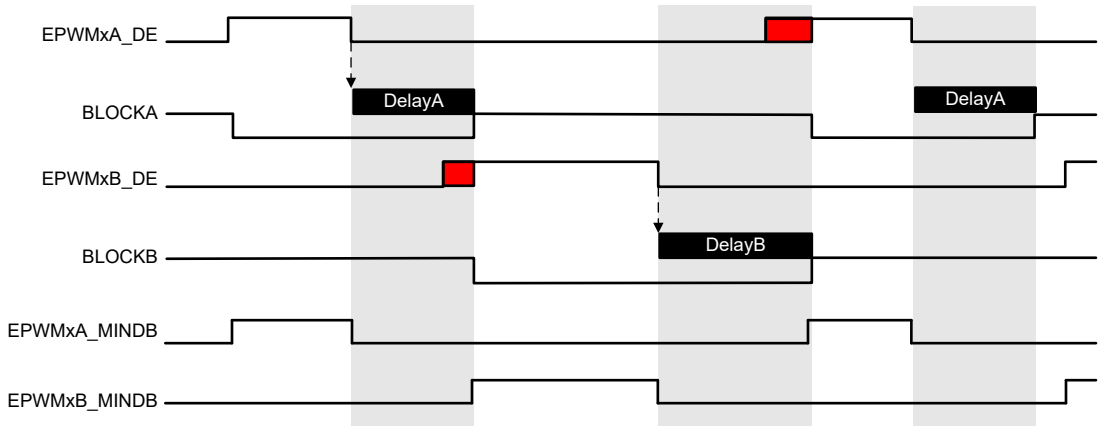


Figure 30-64. Example: Rising Edge on EPWMxA_DE and EPWMxB_DE While Delay is Being Applied

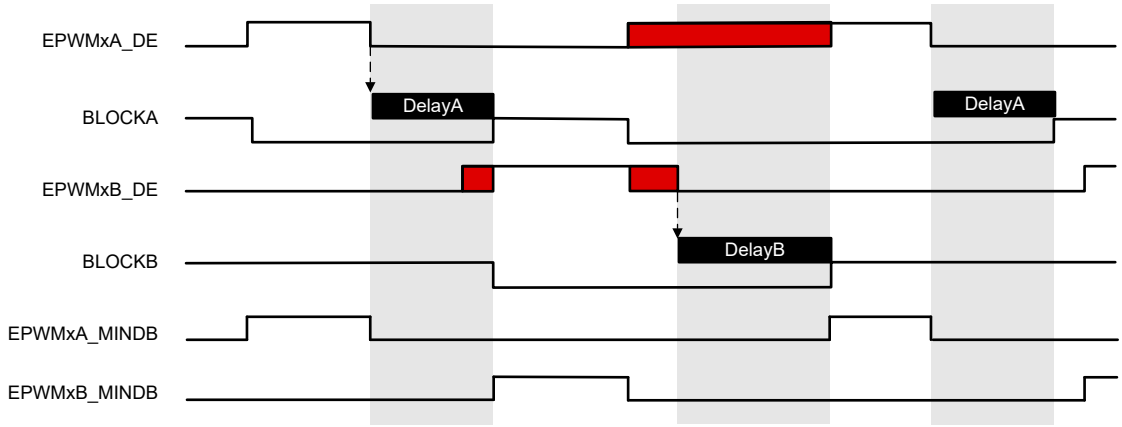


Figure 30-65. Example: Rising Edge on EPWMxA_DE while EPWMxB_DE is Still High

Figure 30-66 illustrates that a rising edge during the delay application does not affect the BLOCKA generation, same behavior is applied to BLOCKB.

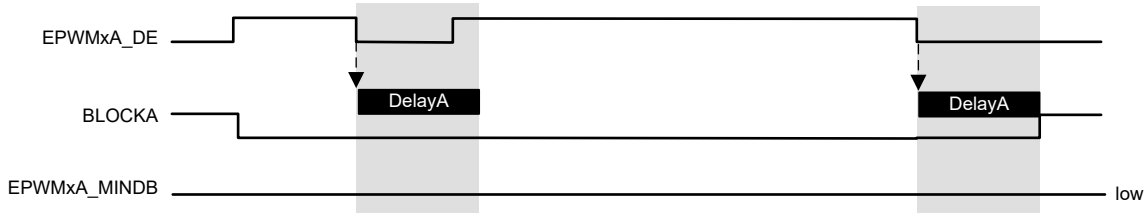


Figure 30-66. Rising Edge During Delay

Figure 30-67 showcases what happens when another falling edge occurs during the delay application. In this scenario, BLOCKA stays low until both DELAYA values are complete, same behavior is applied to BLOCKB.

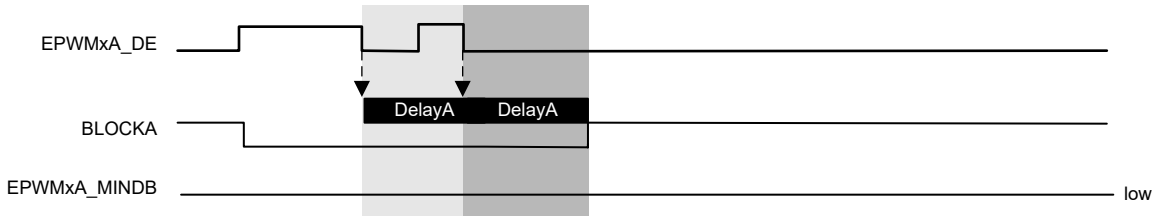


Figure 30-67. Rising Edge and Falling Edge During Delay

30.12.2 Illegal Combo Logic (ICL)

As PWM generation logic gets more configurable and the interaction between multiple PWM instances increases, there is potential for corner cases during applications resulting in unintended PWM states. To detect and make sure that under no circumstance, the PWM states result in potentially hazardous combinations, a Look Up Table (LUT) has been added. By default, the LUT logic is bypassed, LUTXTLx[BYPASS]. When not bypassed, based on the combination of values on Input 3 (IN3), Input 2 (IN2), and Input 1 (IN1), the value driven on OUT is determined by the bits in the LUTCTLx [23:16] register. Input 3 into the LUT comes from one of the ICL X-BAR inputs.

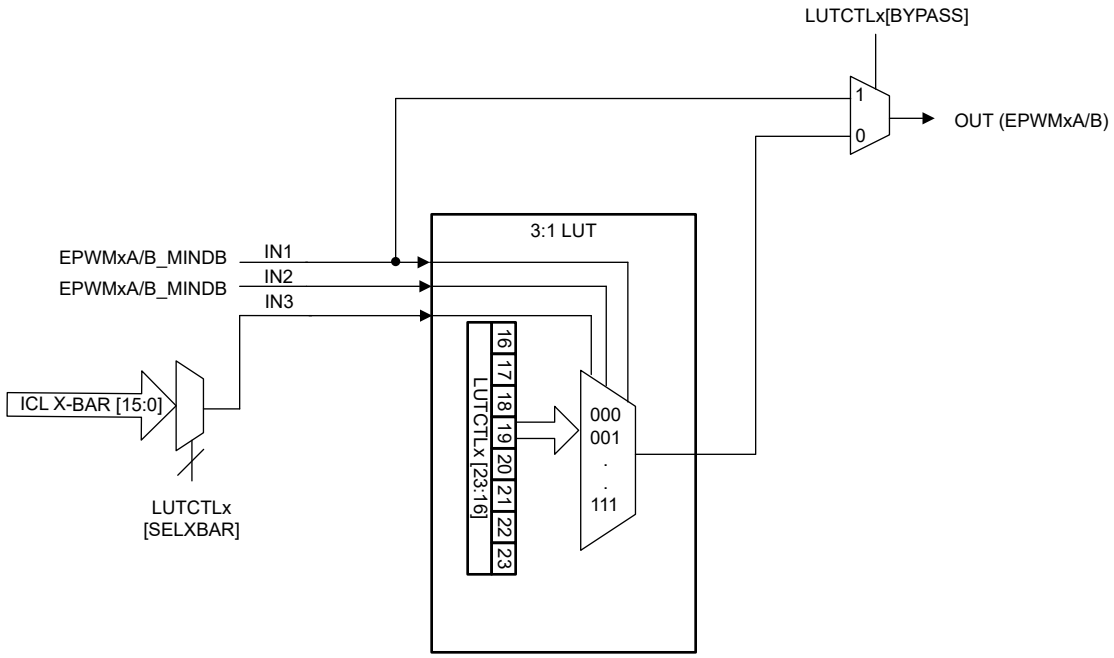


Figure 30-68. Illegal Combo Logic Block Diagram

30.13 Event-Trigger (ET) Submodule

The key functions of the event-trigger submodule are:

- Receives event inputs generated by the time-base, counter-compare, and digital-compare submodules
- Uses the time-base direction information for up/down event qualification
- Uses prescaling logic to issue interrupt requests and ADC start of conversion at:
  - Every event
  - Every second event
  - Up to every fifteenth event
- Provides full visibility of event generation using event counters and flags
- Allows software forcing of Interrupts and ADC start of conversion

The event-trigger submodule manages the events generated by the time-base submodule, the counter-compare submodule, and the digital-compare submodule to generate an interrupt to the CPU and a start of conversion pulse to the ADC when a selected event occurs.

Figure 30-69 illustrates the event-trigger submodule within the ePWM.

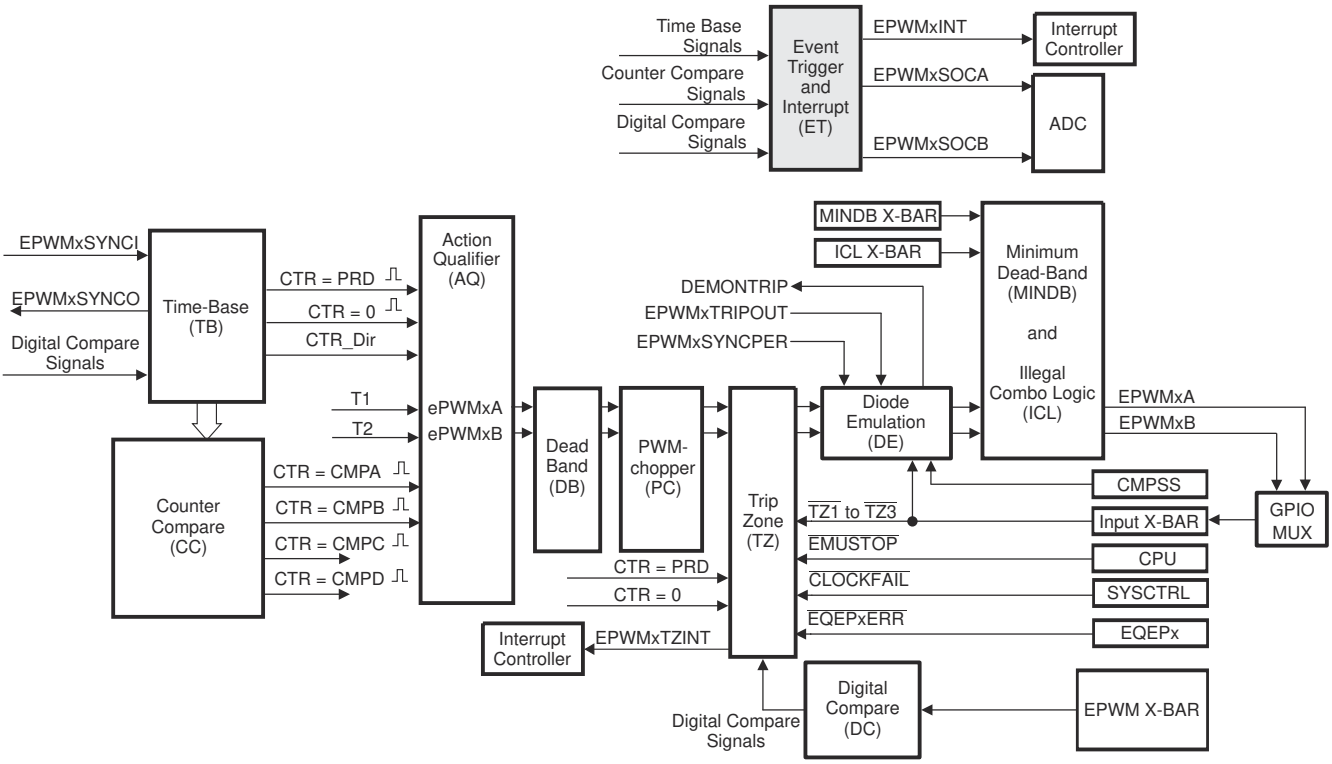


Figure 30-69. Event-Trigger Submodule

30.13.1 Operational Overview of the ePWM Event-Trigger Submodule

The event-trigger submodule monitors various event conditions (shown as inputs on the left side of Figure 30-70) and can be configured to prescale these events before issuing an Interrupt request or an ADC start of conversion. The event-trigger prescaling logic can issue Interrupt requests and ADC start of conversion at:

- Every event
- Every second event
- Up to every fifteenth event

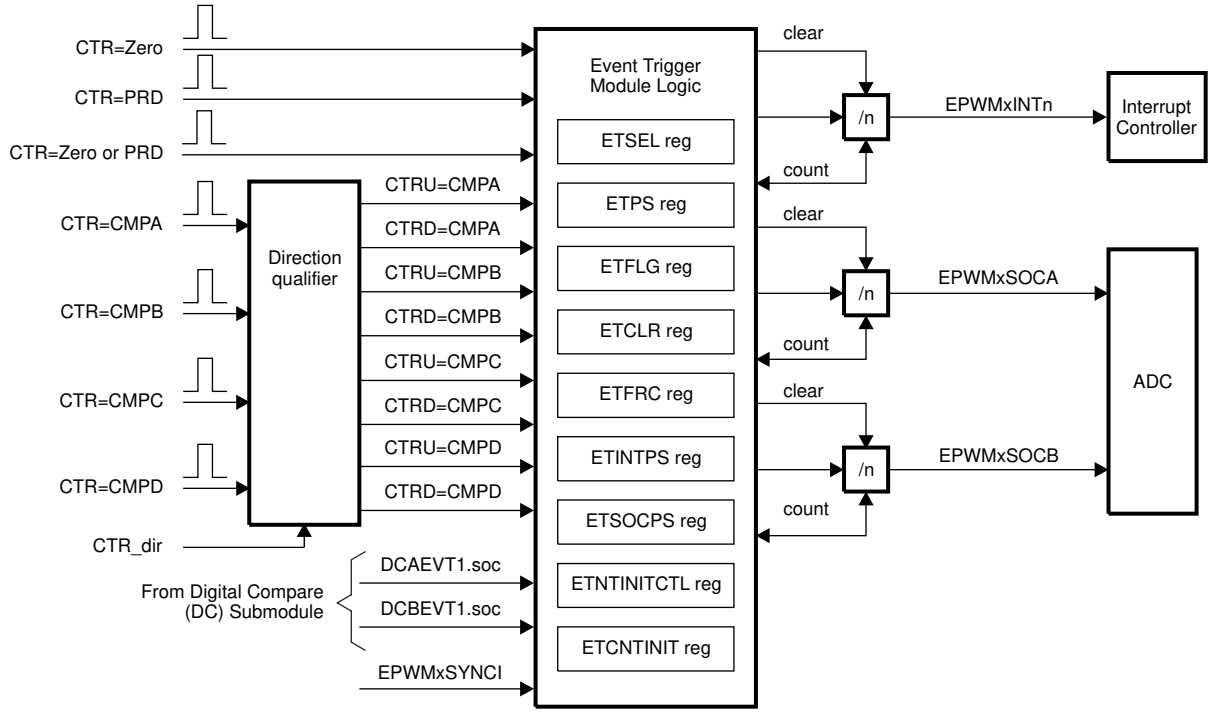


Figure 30-70. Event-Trigger Submodule Showing Event Inputs and Prescaled Outputs

- ETSEL - This selects which of the possible events trigger an interrupt or start an ADC conversion.
- ETPS - This programs the event prescaling options mentioned above.
- ETFLG - These are flag bits indicating status of the selected and prescaled events.
- ETCLR - These bits allow clearing the flag bits in the ETFLG register using software.
- ETFRC - These bits allow software forcing of an event. Useful for debugging or software intervention.
- ETINTPS - This programs the interrupt event prescaling options, supporting count and period up to 15 events.
- ETSOCPS - This programs the SOC event prescaling options, supporting count and period up to 15 events.
- ETCNTINITCTL - These bits enable ETCNTINIT initialization using SYNC event or using software force.
- ETCNTINIT - These bits allow initializing INT/SOCA/SOCB counters on SYNC events (or software force) with user programmed value.

A more detailed look at how the various register bits interact with the Interrupt and ADC start of conversion logic are shown in Figure 30-71, Figure 30-72, and Figure 30-73.

Figure 30-71 shows the event-trigger's interrupt generation logic. The interrupt-period (ETPS[INTPRD]) bits specify the number of events required to cause an interrupt pulse to be generated. The choices available are:

- Do not generate an interrupt.
- Generate an interrupt on every event.
- Generate an interrupt on every second event.
- Generate an interrupt on every third event.

The selection made on ETPS[INTPSSEL] bit determines whether ETINTPS register, INTCNT2 and INTPRD2 bit fields determine frequency of events (interrupt once every 0-15 events).

The event that can cause an interrupt is configured by the interrupt selection (ETSEL[INTSEL]) and (ETSEL[INTSELCMP]) bits. The event can be one of the following:

- Time-base counter equal to zero (TBCTR = 0x00).
- Time-base counter equal to period (TBCTR = TBPRD).
- Time-base counter equal to zero or period (TBCTR = 0x00 || TBCTR = TBPRD).
- Time-base counter equal to the compare A register (CMPA) when the timer is incrementing.
- Time-base counter equal to the compare A register (CMPA) when the timer is decrementing.
- Time-base counter equal to the compare B register (CMPB) when the timer is incrementing.
- Time-base counter equal to the compare B register (CMPB) when the timer is decrementing.
- Time-base counter equal to the compare C register (CMPC) when the timer is incrementing.
- Time-base counter equal to the compare C register (CMPC) when the timer is decrementing.
- Time-base counter equal to the compare D register (CMPD) when the timer is incrementing.
- Time-base counter equal to the compare D register (CMPD) when the timer is decrementing.

The number of events that have occurred can be read from the interrupt event counter ETPS[INTCNT] or ETINTPS[INTCNT2] register bits based off of the selection made using ETPS[INTPSSEL]. That is, when the specified event occurs the ETPS[INTCNT] or ETINTPS[INTCNT2] bits are incremented until the bits reach the value specified by ETPS[INTPRD] or ETINTPS[INTPRD2] determined again by the selection made in ETPS[INTPSSEL]. When ETPS[INTCNT] = ETPS[INTPRD], the counter stops counting and the counter output is set. The counter is only cleared when an interrupt is sent to the interrupt controller.

When ETPS[INTCNT] reaches ETPS[INTPRD], the following behavior occurs. [The following behavior is also applicable to ETINTPS[INTCNT2] and ETINTPS[INTPRD2]:

- If interrupts are enabled, ETSEL[INTEN] = 1 and the interrupt flag is clear, ETFLG[INT] = 0, then an interrupt pulse is generated and the interrupt flag is set, ETFLG[INT] = 1, and the event counter is cleared ETPS[INTCNT] = 0. The counter begins counting events again.
- If interrupts are disabled, ETSEL[INTEN] = 0, or the interrupt flag is set, ETFLG[INT] = 1, the counter stops counting events when the counter reaches the period value ETPS[INTCNT] = ETPS[INTPRD].
- If interrupts are enabled, but the interrupt flag is already set, then the counter holds the output high until the ENTFLG[INT] flag is cleared. This allows for one interrupt to be pending while one is serviced.

Writing a 0 to the INTPRD bits automatically clears the counter (INTCNT = 0) and the counter output resets (so no interrupts are generated). For all other writes to INTPRD, INTCNT retains the previous value. INTCNT resets when INTCNT overflows. Writing a 1 to the ETFRC[INT] bit increments the event counter INTCNT. The counter behaves as previously described when INTCNT = INTPRD. When INTPRD = 0, the counter is disabled and hence no events are detected and the ETFRC[INT] bit is also ignored. The same applies to ETINTPS[INTCNT2] and ETINTPS[INTPRD2].

The previous definition means that an interrupt on every event, on every second event, or on every third event if using the INTCNT and INTPRD can be generated. An interrupt on every event up to 15 events if using the INTCNT2 and INTPRD2 can be generated.

The INTCNT2 value can be initialized with the value from ETCNTINIT[INTINIT] based on the selection made in ETCNTINITCTL[INTINITEN]. When ETCNTINITCTL[INTINITEN] is set, then initialization of INTCNT2 counter with contents of ETCNTINIT[INTINIT] on a SYNC event or software force is determined by ETCNTINITCTL[INTINITFRC].

ETINTMIX, ETSOCAMIX and ETSOCBMIX Signals

In type 5 ePWM, the Event-Trigger submodule can generate and use ETINTMIX, ETSOCAMIX and ETSOCBMIX signals.

- **ETINTMIX**: This signal is a generated from the ORed combination of the sources enabled in the ETINTMIXEN register. The ETINTMIX signal can be used as a source for the EPWMxINT interrupt.
- **ETSOCAMIX**: This signal is a generated from the ORed combination of the sources enabled in the ETSOCAMIXEN register. The ETSOCAMIX signal can be used as a source for the EPWMxSOCA trigger signal.
- **ETSOCBMIX**: This signal is a generated from the ORed combination of the sources enabled in the ETSOCBMIXEN register. The ETSOCBMIX signal can be used as a source for the EPWMxSOCB trigger signal

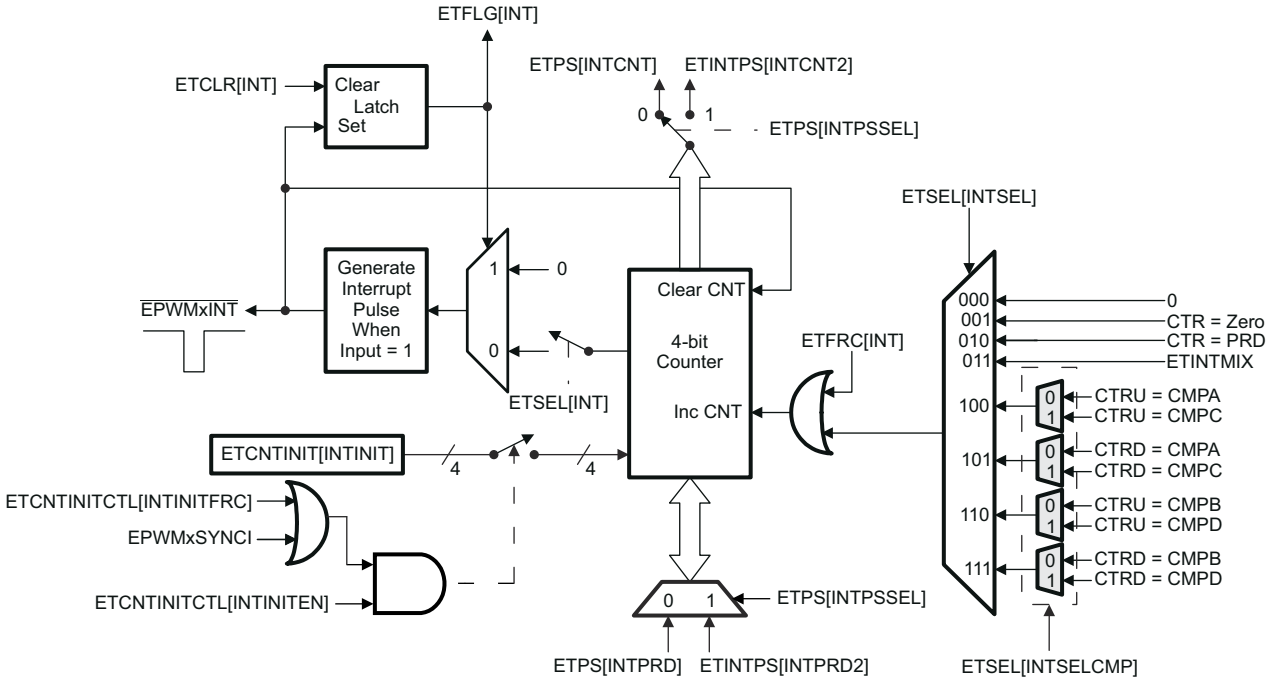
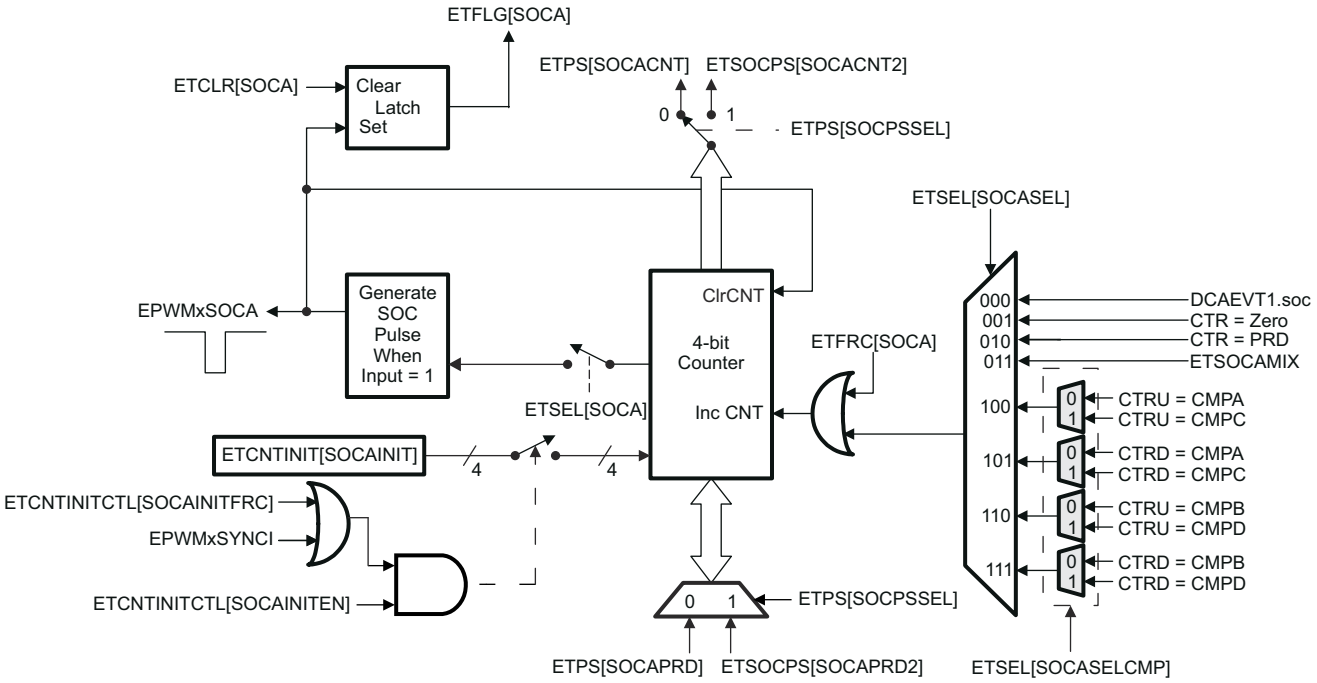


Figure 30-71. Event-Trigger Interrupt Generator

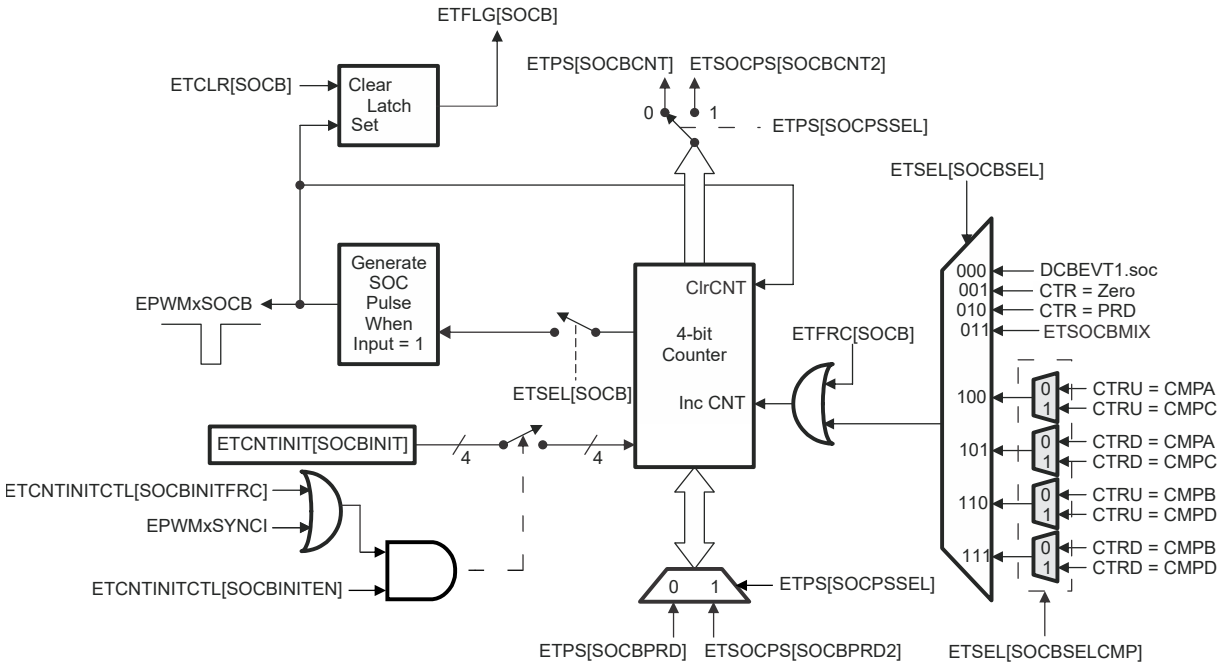
Figure 30-72 shows the operation of the event-trigger's start-of-conversion-A (SOCA) pulse generator. The enhancements include SOCASELCMP and SOCBSELCMP bit fields defined in the ETSEL register enable CMPC and CMPD events respectively to cause a start of conversion. The ETPS[SOCPSSEL] bit field determines whether SOCACNT2 and SOCAPRD2 take control or not. The ETPS[SOCACNT] counter and ETPS[SOCAPRD] period values behave similarly to the interrupt generator except that the pulses are continuously generated. That is, the pulse flag ETFLG[SOCA] is latched when a pulse is generated, but the interrupt generator does not stop further pulse generation. The enable and disable bit ETSEL[SOCAEN] stops pulse generation, but input events can still be counted until the period value is reached as with the interrupt generation logic. The event that triggers an SOCA and SOCB pulse can be configured separately in the ETSEL[SOCASEL] and ETSEL[SOCBSEL] bits. The possible events are the same events that can be specified for the interrupt generation logic with the addition of the DCAEVT1.soc and DCBEVT1.soc event signals from the digital compare (DC) submodule. The SOCACNT2 initialization scheme is very similar to the interrupt generator with respective enable, value initialize and SYNC or software force options.



NOTE: The DCAEVT1.soc signals are generated by the Digital Compare (DC) submodule in Section 30.14.

Figure 30-72. Event-Trigger SOCA Pulse Generator

Figure 30-73 shows the operation of the event-trigger's start-of-conversion-B (SOCB) pulse generator. The event-trigger's SOCB pulse generator operates the same way as the SOCA.



NOTE: The DCBEVT1.soc signals are generated by the Digital Compare (DC) submodule in Section 30.14.

Figure 30-73. Event-Trigger SOCB Pulse Generator

30.14 Digital Compare (DC) Submodule

Figure 30-74 illustrates where the digital compare (DC) submodule signals interface to other submodules in the ePWM system.

The eCAP input signals are sourced from the Input X-BAR signals as shown in Figure 30-75.

On this device, any of the GPIO pins can be flexibly mapped to be the trip-zone input and trip inputs to the trip-zone submodule and digital compare submodule. The Input X-BAR Input Select (INPUTxSELECT) register defines which GPIO pins gets assigned to be the trip-zone inputs / trip inputs.

The digital compare (DC) submodule compares signals external to the ePWM module (for instance, CMPSSx signals from the analog comparators) to directly generate PWM events/actions which then feed to the event-trigger, trip-zone, and time-base submodules. Additionally, blanking window functionality is supported to filter noise or unwanted pulses from the DC event signals.

Note

The user is responsible for driving the correct state on the selected pin before enabling the clock and configuring the trip input for the respective ePWM peripheral to avoid spurious latch of the TRIP signal.

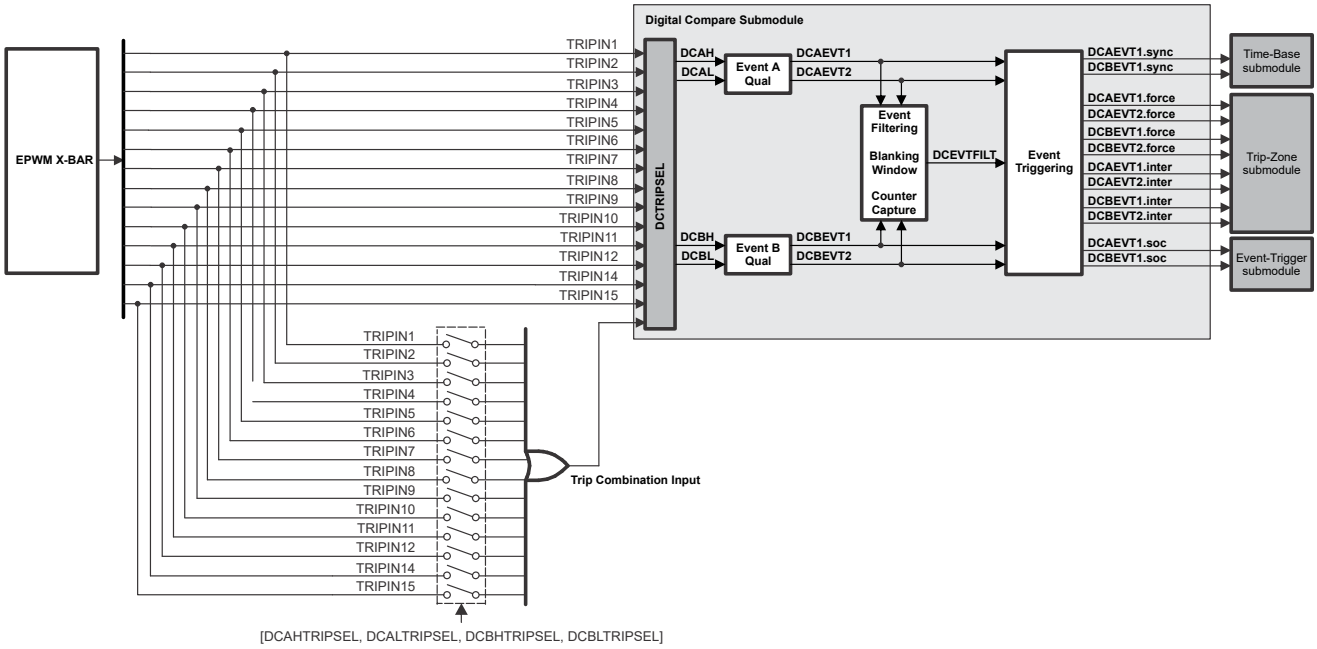


Figure 30-74. Digital-Compare Submodule High-Level Block Diagram

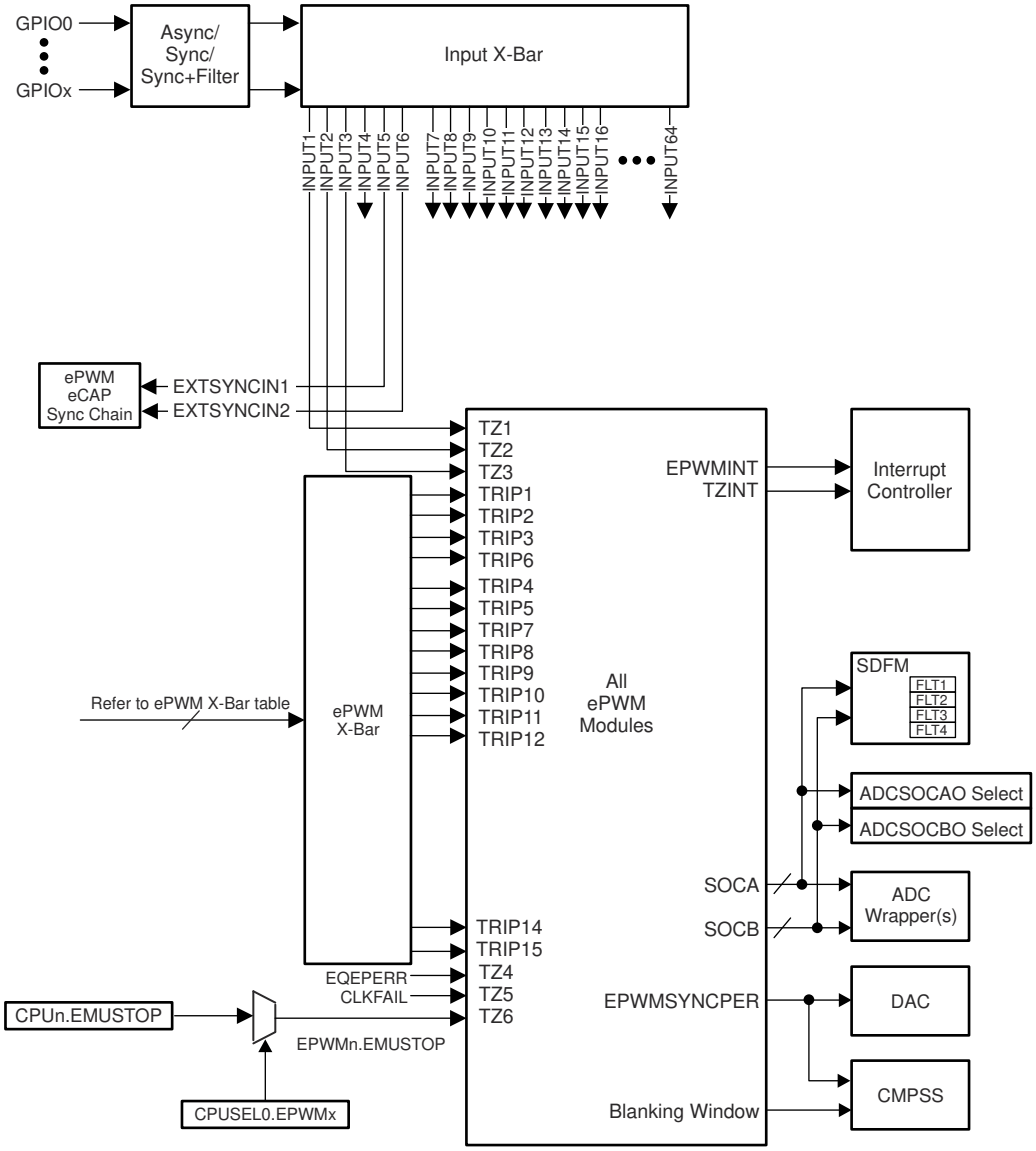


Figure 30-75. GPIO MUX-to-Trip Input Connectivity

30.14.1 Purpose of the Digital Compare Submodule

The key functions of the digital compare submodule are:

- Analog comparator (COMP) module outputs fed though the Input X-BAR, EPWM X-BAR, externally using the GPIO peripheral, interrupt controller signals, ECC error signals, TZ1, $\overline{TZ2}$, and $\overline{TZ3}$ inputs generate Digital Compare A High/Low (DCAH, DCAL) and Digital Compare B High/Low (DCBH, DCBL) signals.
- DCAH/L and DCBH/L signals trigger events that can then either be filtered or applied directly to the trip-zone, event-trigger, and time-base submodules to:
  - generate a trip zone interrupt
  - generate an ADC start of conversion
  - force an event
  - generate a synchronization event for synchronizing the ePWM module TBCTR.
- Event filtering (blanking window logic) can optionally blank the input signal to remove noise.

30.14.2 Enhanced Trip Action Using CMPSS

To allow multiple CMPSS at a time to affect DCA/BEVTx events and trip actions, there is a OR logic to bring together ALL trip inputs (up to 15) from sources external to the ePWM module and feed into DCAH, DCAL, DCBH, and DCBL as a “combinational input” using the DCTRIPSEL register. This is configured by selecting “Trip combination input” (value of 0xF) in the DCTRIPSEL register.

There is a discrete choice of which trip inputs to put through the combinational logic for generating the DCAH, DCAL, DCBH, and DCBL signals. This is achieved using the DCAHTRIPSEL, DCALTRIPSEL, DCBHTRIPSEL, and DCBLTRIPSEL register selections. Inputs selected for combinational input are passed through to the DCTRIPSEL register.

30.14.3 Using CMPSS to Trip the ePWM on a Cycle-by-Cycle Basis

When using the CMPSS to trip the ePWM on a cycle-by-cycle basis, steps can be taken to prevent an asserted comparator trip state in one PWM cycle from extending into the following cycle. The CMPSS can be used to signal a trip condition to the downstream ePWM modules. For applications like peak current mode control, only one trip event per PWM cycle is expected. Under certain conditions, it is possible for a sustained or late trip event (arriving near the end of a PWM cycle) to carry over into the next PWM cycle if precautions are not taken. If either the CMPSS Digital Filter or the ePWM Digital Compare (DC) submodule is configured to qualify the comparator trip signal, “N” number of clock cycles of qualification are introduced before the ePWM trip logic can respond to logic changes of the trip signal. Once an ePWM trip condition is qualified, the trip condition remains active for N clock cycles after the comparator trip signal has de-asserted. If a qualified comparator trip signal remains asserted within N clock cycles prior to the end of a PWM cycle, the trip condition is not cleared until after the following PWM cycle has started. Thus, the new PWM cycle detects a trip condition as soon as the cycle begins.

To avoid this undesired trip condition, the application can take steps to make sure that the qualified trip signal seen by the ePWM trip logic is deasserted prior to the end of each PWM cycle. This can be accomplished through various methods:

- Design the system such that a comparator trip is not asserted within N clock cycles prior to the end of the PWM cycle.
- Activate blanking of the comparator trip signal using the ePWM event filter at least two clock cycles prior to the PWMSYNCPER signal and continue blanking for at least N clock cycles into the next PWM cycle.
- If the CMPSS COMPxLATCH path is used, clear the COMPxLATCH at least N clock cycles prior to the end of the PWM cycle. The latch can be cleared by software (using COMPSTSCLR) or by generating an early PWMSYNCPER signal. The ePWM modules on this device include the ability to generate PWMSYNCPER upon a CMPC or CMPD match (using HRPCTL) for arbitrary PWMSYNCPER placement within the PWM cycle.

30.14.4 Operation Highlights of the Digital Compare Submodule

The following sections describe the operational highlights and configuration options for the digital compare submodule.

30.14.4.1 Digital Compare Events

As described in Section 30.14.1, trip zone inputs ($\overline{TZ1}$, $\overline{TZ2}$, and $\overline{TZ3}$) and CMPSSx signals from the analog comparator (COMP) module can be selected using the DCTRIPSEL bits to generate the Digital Compare A High and Low (DCAH/L) and Digital Compare B High and Low (DCBH/L) signals. Then, the configuration of the TZDCSEL register qualifies the actions on the selected DCAH/L and DCBH/L signals, which generate the DCAEVT1/2 and DCBEVT1/2 events (Event Qualification A and B).

Note

The $\overline{TZn}$ signals, when used as a DCEVT tripping functions, are treated as a normal input signal and can be defined to be active-high or active-low inputs. ePWM outputs are asynchronously tripped when either the $\overline{TZn}$, DCAEVTx.force, or DCBEVTx.force signals are active. For the condition to remain latched, a minimum of 3 × TBCLK sync pulse width is required. If pulse width is < 3*TBCLK sync pulse width, the trip condition can or can not get latched by CBC or OST latches.

The DCAEVT1/2 and DCBEVT1/2 events can then be filtered to provide a filtered version of the event signals (DCEVTFILT) or the filtering can be bypassed. Filtering is discussed further in Event Filtering. Either the DCAEVT1/2 and DCBEVT1/2 event signals or the filtered DCEVTFILT event signals can generate a force to the trip zone module, a TZ interrupt, an ADC SOC, or a PWM sync signal.

- **force signal:** DCAEVT1/2.force signals force trip zone conditions which either directly influence the output on the EPWMxA pin (using TZCTL, TZCTLDCA, TZCTLDCB register configurations) or, if the DCAEVT1/2 signals are selected as one-shot or cycle-by-cycle trip sources (using the TZSEL register), the DCAEVT1/2.force signals can effect the trip action using the TZCTL or TZCTL2 register configurations. The DCBEVT1/2.force signals behaves similarly, but affect the EPWMxB output pin instead of the EPWMxA output pin.

  The priority of conflicting actions on the TZCTL, TZCTL2, TZCTLDCA and TZCTLDCB registers is as follows (highest priority overrides lower priority):

  Output EPWMxA:
  - TZA (highest) -> DCAEVT1 -> DCAEVT2 (lowest)
  - TZAU (highest) -> DCAEVT1U -> DCAEVT2U (lowest)
  - TZAD (highest) -> DCAEVT1D -> DCAEVT2D (lowest)

  Output EPWMxB:
  - TZB (highest) -> DCBEVT1 -> DCBEVT2 (lowest)
  - TZBU (highest) -> DCBEVT1U -> DCBEVT2U (lowest)
  - TZBD (highest) -> DCBEVT1D -> DCBEVT2D (lowest)
- **interrupt signal:** DCAEVT1/2.interrupt signals generate trip zone interrupts to the interrupt controller. To enable the interrupt, set the DCAEVT1, DCAEVT2, DCBEVT1, or DCBEVT2 bits in the TZEINT register. Once one of these events occurs, an EPWMxTZINT interrupt is triggered, and the corresponding bit in the TZCLR register must be set to clear the interrupt.
- **soc signal:** The DCAEVT1.soc signal interfaces with the event-trigger submodule and can be selected as an event which generates an ADC start-of-conversion-A (SOCA) pulse using the ETSEL[SOCASEL] bit. Likewise, the DCBEVT1.soc signal can be selected as an event which generates an ADC start-of-conversion-B (SOCB) pulse using the ETSEL[SOCBSEL] bit.
- **sync signal:** The DCAEVT1.sync and DCBEVT1.sync events are ORed with the EPWMxSYNCI input signal and the TBCTL[SWFSYNC] signal to generate a synchronization pulse to the time-base counter.

Figure 30-76 and Figure 30-77 show how the DCxEVT1, DCxEVT2, or DCEVTFLT signals are processed to generate the digital compare A and B event force, interrupt, soc and sync signals.

In some of the applications like Phase Shifted Full Bridge (PSFB) Converters, it is required that different actions are taken on a CBC trip event and an OST trip event. This can be achieved using the DCxEVT1LAT.

- This latch can be cleared on CNT = 0, CTR = PRD, and CNT = 0 OR CTR = PRD events based on the setting of DCxCTL.EVTy.LATCLRSEL setting. This is similar to CBC latch clear mechanism.
- DCxEVTy.force signal can be chosen to be either the latched version or the unlatched version based on DCxCTL.EVTyLATSEL value.
- The status of DCxEVTyLAT signal can be accessed by reading DCxCTL.EVTyLAT field.

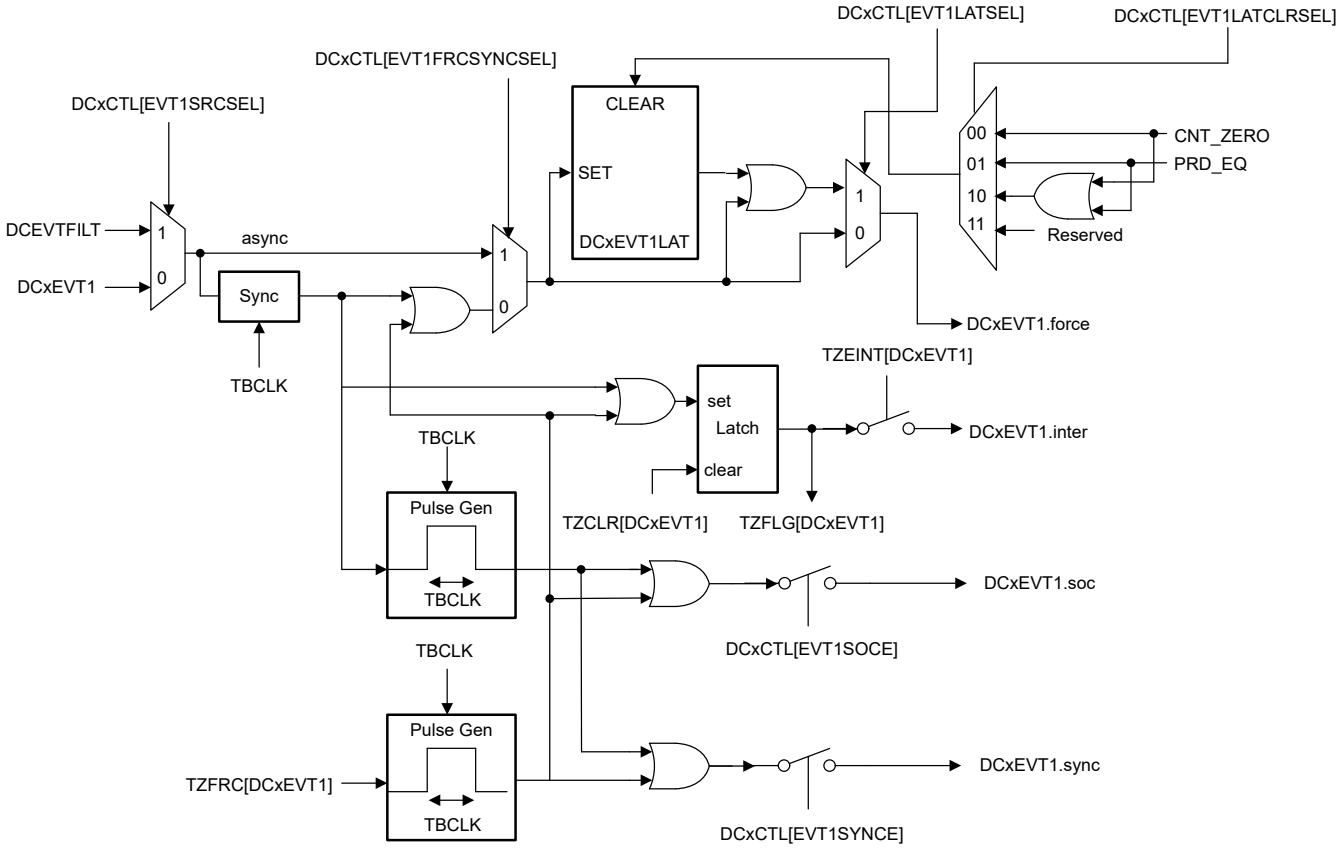


Figure 30-76. DCxEVT1 Event Triggering

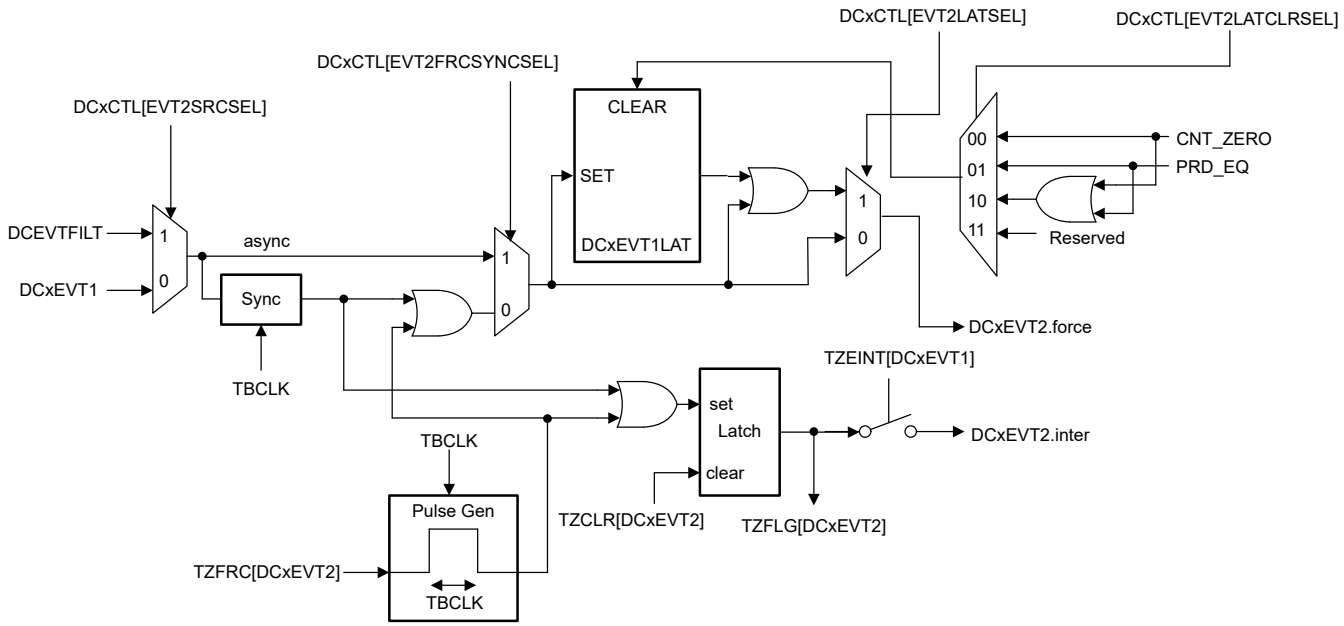


Figure 30-77. DCxEVT2 Event Triggering

30.14.4.2 Event Filtering

Blank Control Logic: The DCAEVT1/2 and DCBEVT1/2 events can be filtered using event filtering logic to remove noise by optionally blanking events for a certain period of time. This is useful for cases where the analog comparator outputs can be selected to trigger DCAEVT1/2 and DCBEVT1/2 events, and the blank control logic is used to filter out potential noise on the signal prior to tripping the PWM outputs or generating an interrupt or ADC start-of-conversion. Blank control logic is used to define a blanking window, which ignores all event occurrences on the signal while the window is active. The blanking window is configured in the DCFCTL, DCFOFFSET, and DCFWINDOW registers. The DCFCTL register enables the blanking window and aligns the blanking window to either a CTR = PRD pulse or a CTR = 0 pulse or both CTR = PRD and CTR = 0 as specified by DCFCTL[PULSESEL]. DCFCTL[SRCSEL] selects the DCxEVTy event source for the DCEVTFILT signal. An offset value in TBCLK counts is programmed into the DCFOFFSET register, which determines at what point after the CTR = PRD or CTR = 0 pulse the blanking window starts. The duration of the blanking window, in number of TBCLK counts after the offset counter expires, is written to the DCFWINDOW register by the application. Before and after the blanking window ends, events can generate soc, sync, interrupt, and force signals as before. Figure 30-78 shows the details of the event filtering logic.

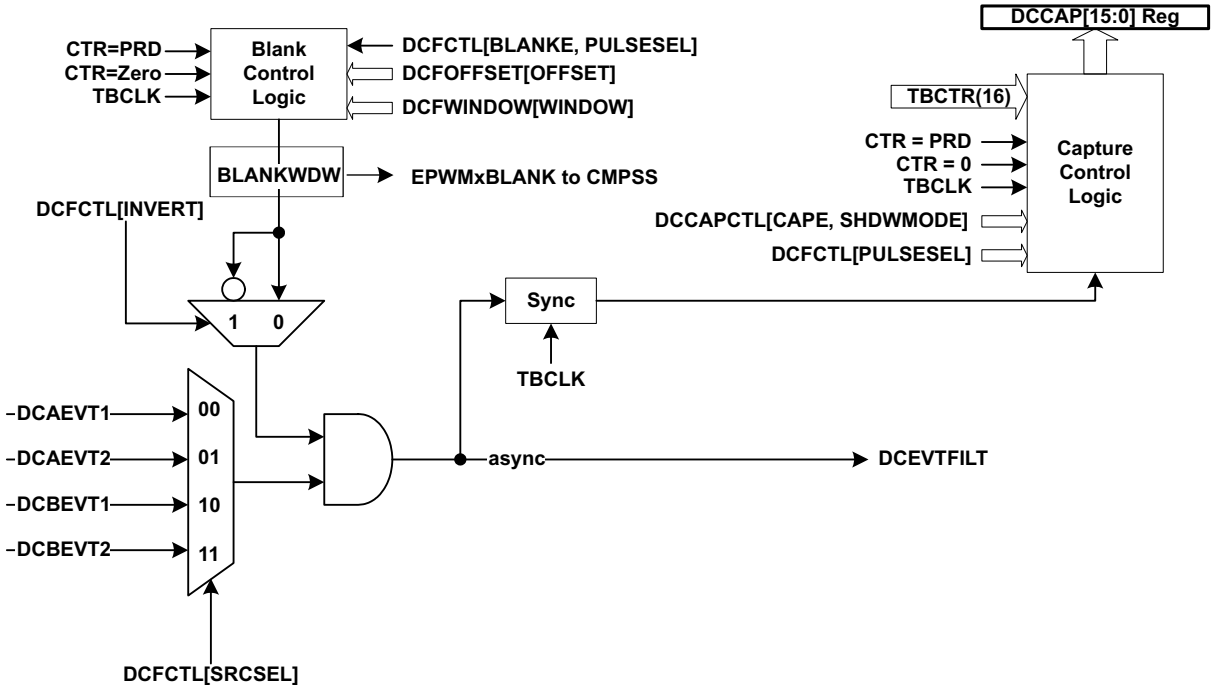


Figure 30-78. Event Filtering

Capture Control Logic: The event filtering can also capture the TBCTR value of the selected DCxEVTy event as configured in the DCCAPCTL register. When capture control logic is enabled, the selected DCxEVTy event triggers capture of the TBCTR to the active register. The CPU reads directly from the active register unless shadow mode is enabled by DCCAPCTL[SHDWMODE]. When shadow mode is enabled, the active register information is copied to shadow register on the event specified by DCFCCTL[PULSESEL], and the CPU reads from the shadow register. After the selected DCxEVTy event, no further capture events occur until the event specified by DCCAPCTL[CAPMODE]. The CAPMODE can be configured two ways: (1) no further capture events occur until the event defined by DCFCTL[PULSESEL] or (2) no further capture events occur until the compare-event flag at DCCAPCTL[CAPSTS] is cleared by DCCAPCTL[CAPCLR].

Note

You must configure the ePWM blanking window appropriately so that the Trip Input stays valid for at least 3 ePWM cycles after the blanking window has expired.

Figure 30-79 illustrates several timing conditions for the offset and blanking window within an ePWM period. Notice that if the blanking window crosses the CTR = 0 or CTR = PRD boundary, the next window still starts at the same offset value after the CTR = 0 or CTR = PRD pulse.

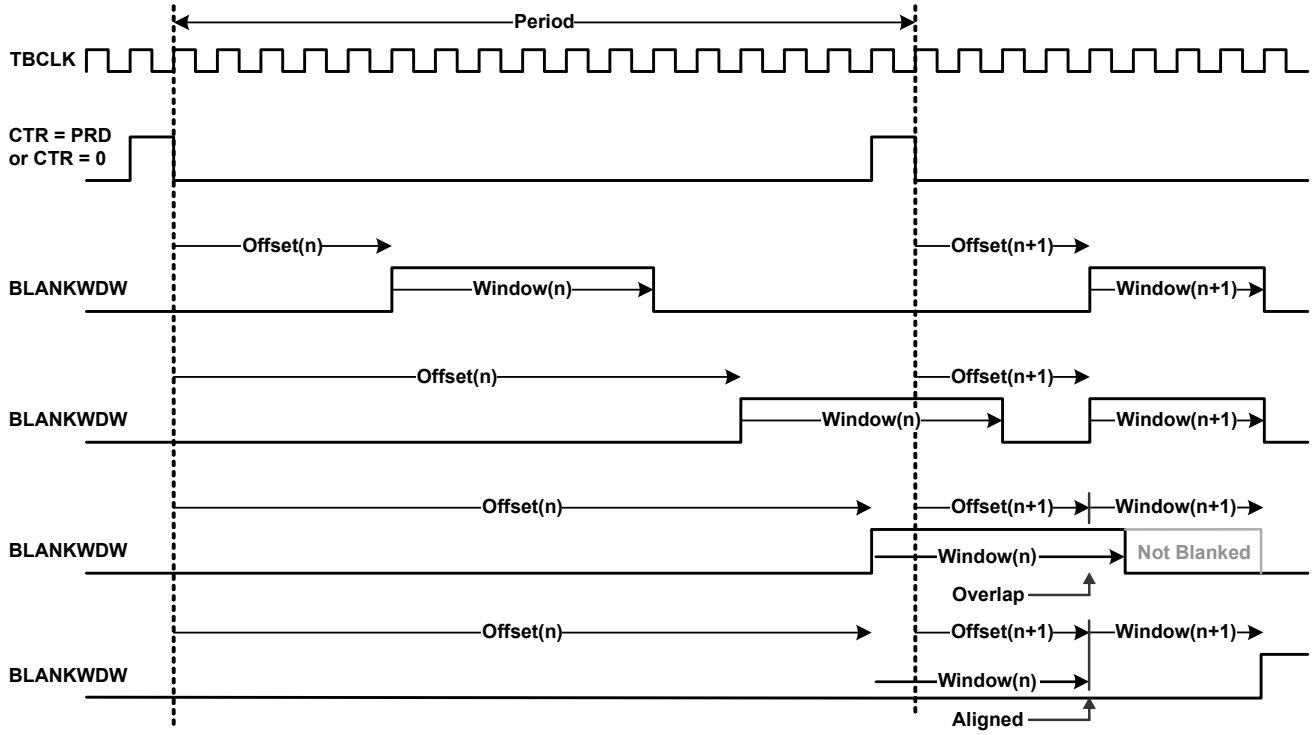


Figure 30-79. Blanking Window Timing Diagram

BLANKPULSEMIX and DCCAPMIX Signals

The CAPCTL MUX (available in the Capture Control Logic) and DCFCTL MUX (available for Blank Control Logic and Capture Control Logic) have new options in type 5 ePWM which allows them to select the DCCAPMIX or BLANKPULSEMIX signal respectively.

In type 5 ePWM, the shadow load signal for the Capture Control Logic can be different from the blanking window alignment signal (which is selected by DCFCTL[PULSESEL]). The CAPCTL mux can be configured to use the DCCAPMIX signal

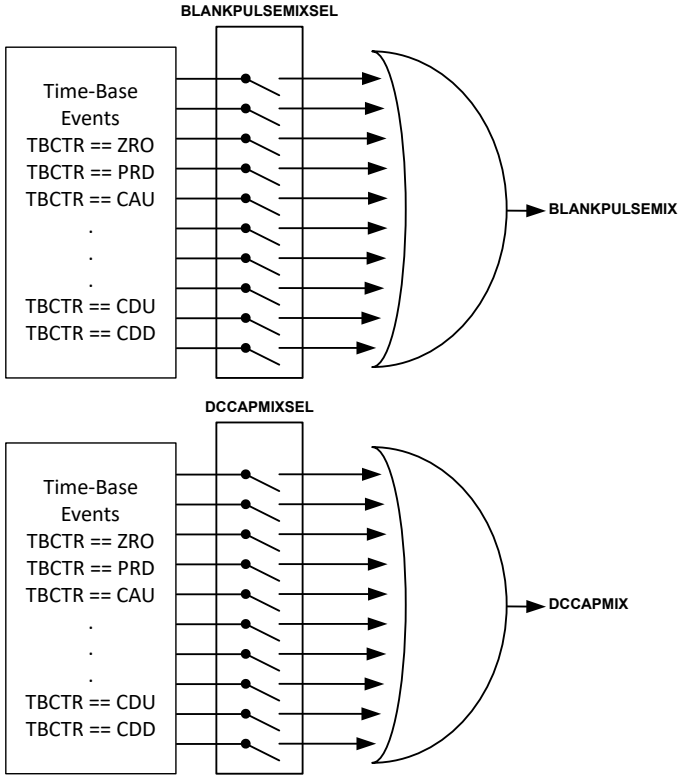


Figure 30-80. BLANKPULSEMIX and DCCAPMIX Signal Source

30.14.4.3 Valley Switching

Event filtering depicts the valley switching function along with the event filtering logic described in Event Filtering. This function can be used to achieve programmable valley switching without any additional external circuitry. This module provides an on-chip hardware mechanism that can:

- Capture the oscillation period
- Accurately delay the PWM switching instant
- Allow a programmable number of edges before the delay takes effect
- Provide multiple choices of triggers and events
- Allow easy adaptability for optimum performance under changing system/operating conditions

The DCxEVTy signal needs further processing to support valley switching. Here is a brief description of how valley switching function is enabled:

1. Select one of the DCxEVTy events as input to the valley switching block (DCFCTL[SRCSEL]) with an option to add the blanking window (Blank Control Logic). This is where the comparator output (or external input) above is selected as an input to the valley switching block.
2. Configure the edge filter to capture ‘n’ rising, falling or both edges through the edge selection logic (DCFCTL[EDGEMODE, EDGECOUNT]).
3. Select the correct event to reset and restart the edge filter (VCAPCTL[TRIGSEL]). Edge capturing event is triggered or armed by this selected edge.
4. Enable valley capture logic (VCAPCTL[VCAPE]).
5. Select the start edge that indicates the start of capture for oscillation period measurement (VCNTCFG[STARTEDGE]). This is where the 16-bit counter starts counting.
6. Select the stop edge (VCNTCFG[STOPEDGE]) that indicates the edge at which the 16-bit counter stops counting. The captured counter value (CNTVAL) provides oscillation period information.
   - The STOPEDGE value must always be greater than STARTEDGE value.
7. Configure and apply the captured delay (CNTVAL) to the edge filtered DCxEVTy signal. The CNTVAL value can be applied as is or applied in conjunction with a software programmed value (useful for offset adjustment) (SWVDELVAL) or only a fraction of the delay can be applied with or without SWVDELVAL. This is useful to correctly apply a delay corresponding to the valley point. (VCAPCTL[VDELAYDIV])
8. Configure VCAPCTL[EDGEFILTDLYSEL] to apply hardware delay based on the captured value above.

Once the counter is stopped, counter value is copied into CNTVAL register and counter is reset to zero. No further captures are done until the logic is triggered again by occurrence of event selected by VCAPCTL[TRIGSEL]. In this implementation, the software trigger is used as the source for VCAPCTL[TRIGSEL]. Upon occurrence of the trigger event, irrespective of the current status of the counter, the counter is reset and starts counting from zero upon occurrence of the STARTEDGE. Similarly, upon occurrence of the trigger event, the edge filter is reset and starts counting from zero upon occurrence of the STARTEDGE.

Output from the valley switching block (DCEVTFILT) is then used to synchronize the PWM time-base. The process is shown in Figure 30-81.

Note

A specific application example showcasing the usage of valley switching hardware and software is available in C2000Ware.

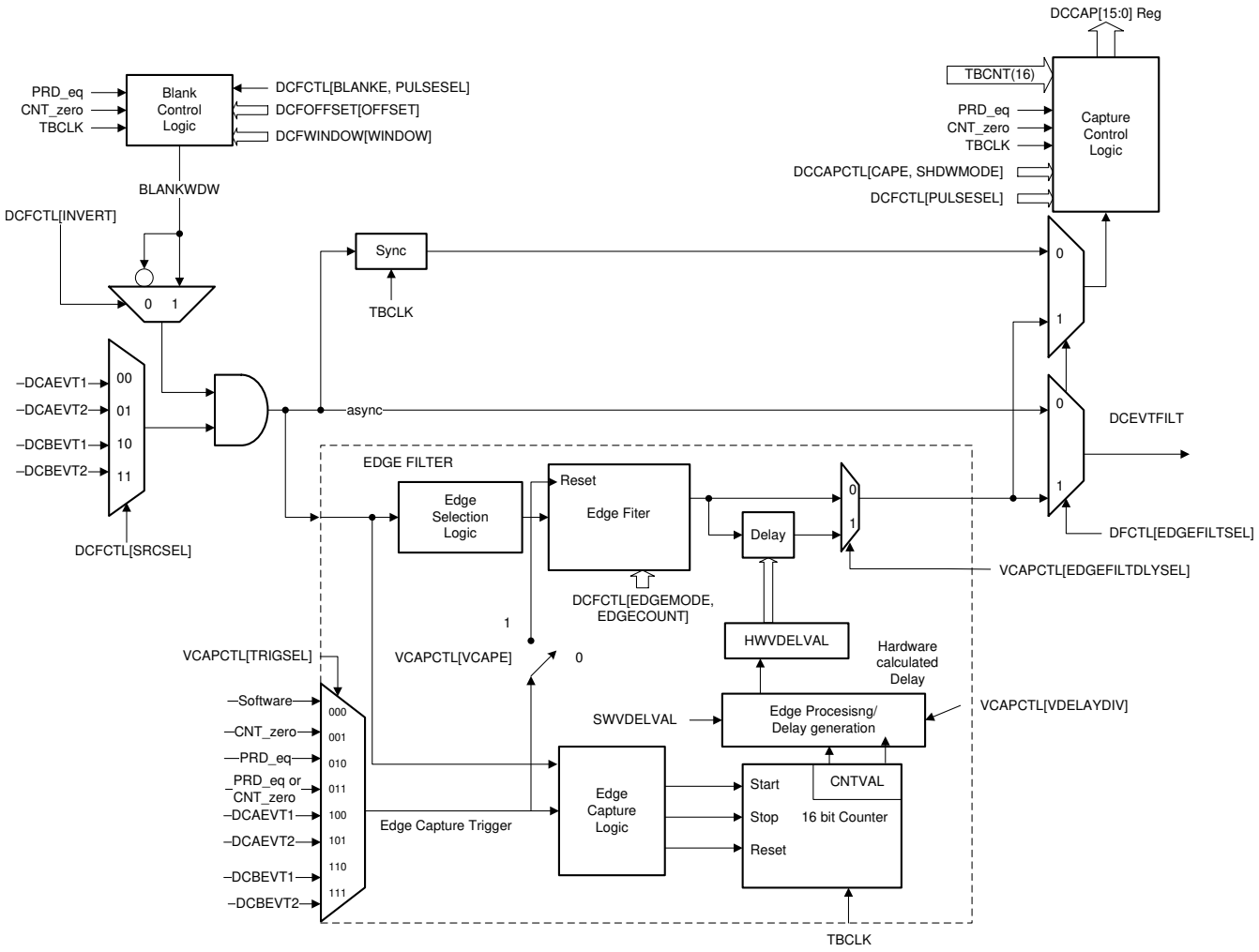


Figure 30-81. Valley Switching

30.14.4.4 Event Detection

This logic is primarily intended to detect an occurrence of a trip event in a configured time window. The window is configured by MIN and MAX values configured in the XMINMAX register sets.

Figure 30-82 indicates the window spread across MIN and MAX bounds and the edge of the chosen signal occurring in that window. The purpose of this block is to detect the occurrence of such edge. If no such edge occurs, this module generates a trip event as well as an interrupt, if configured.

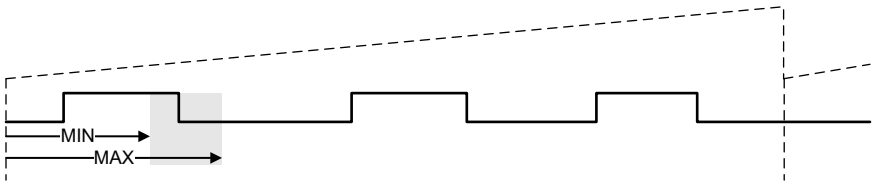


Figure 30-82. MIN, MAX Settings and Window for Capture Event Detection

30.14.4.4.1 Input Signal Detection

The CAPTRIPSEL, CAPINTRIPSEL and CAPGATETRIPSEL muxes are used for signal selection. Figure 30-83 shows how the CAPIN and CAPGATE signal source is selected.

CAPIN Input: This signal (any input coming from EPWM X-BAR) can be configured as the signal input on which the edge detection logic is performed.

CAPGATE Input: This signal (any input coming from EPWM X-BAR) can be configured as the gating signal to Min/Max logic. This signal gates the CAPIN input signal.

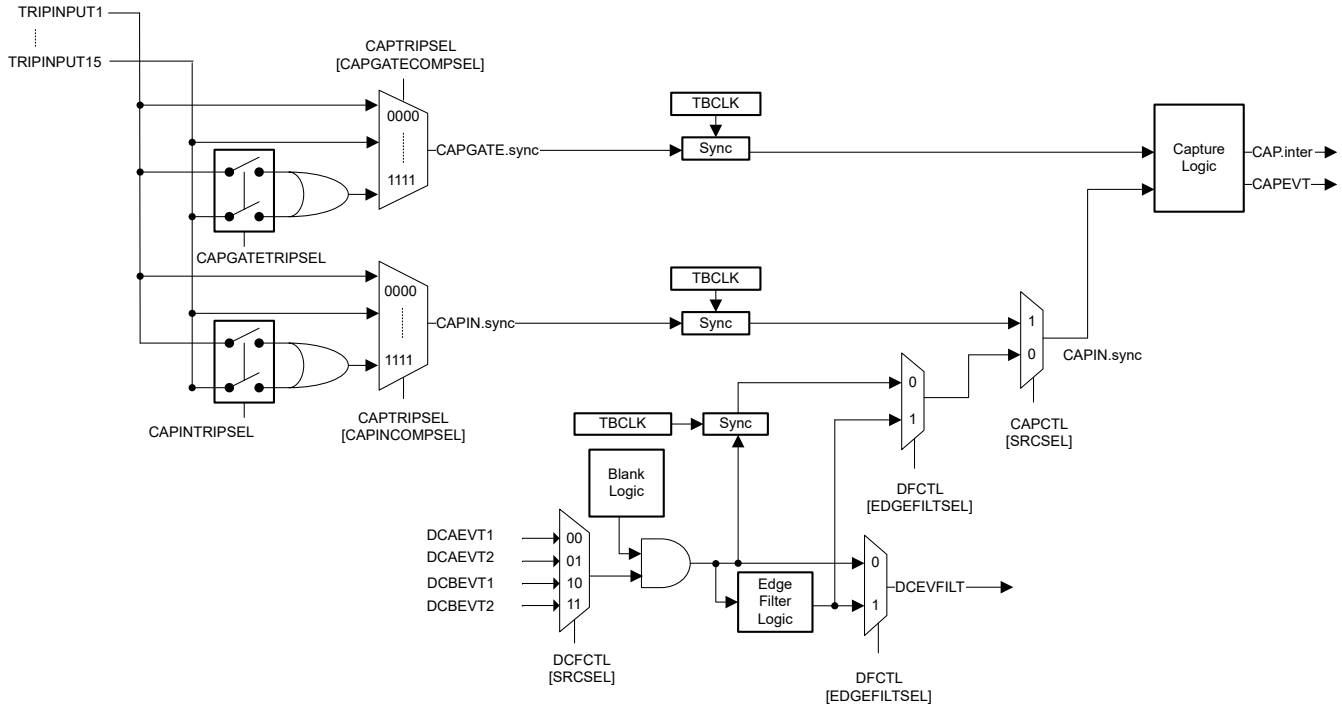


Figure 30-83. CAPIN and CAPGATE Source Selection

Once selected, Figure 30-84 demonstrates how the CAPGATE and CAPIN signals propagate into the counter capture logic. The logic works in the following way:

- CAPCTL[GATEPOL] is used for the polarity selection of the gating input to be optionally inverted or tied to a 0 or 1.
- CAPCTL[CAPINPOL] can be used to select the edge polarity of CAPIN.sync signal. CAPIN.sync signal is selected from the DCEVTFILT options and the CAPIN signal using CAPCTL[SRCSEL] bits.

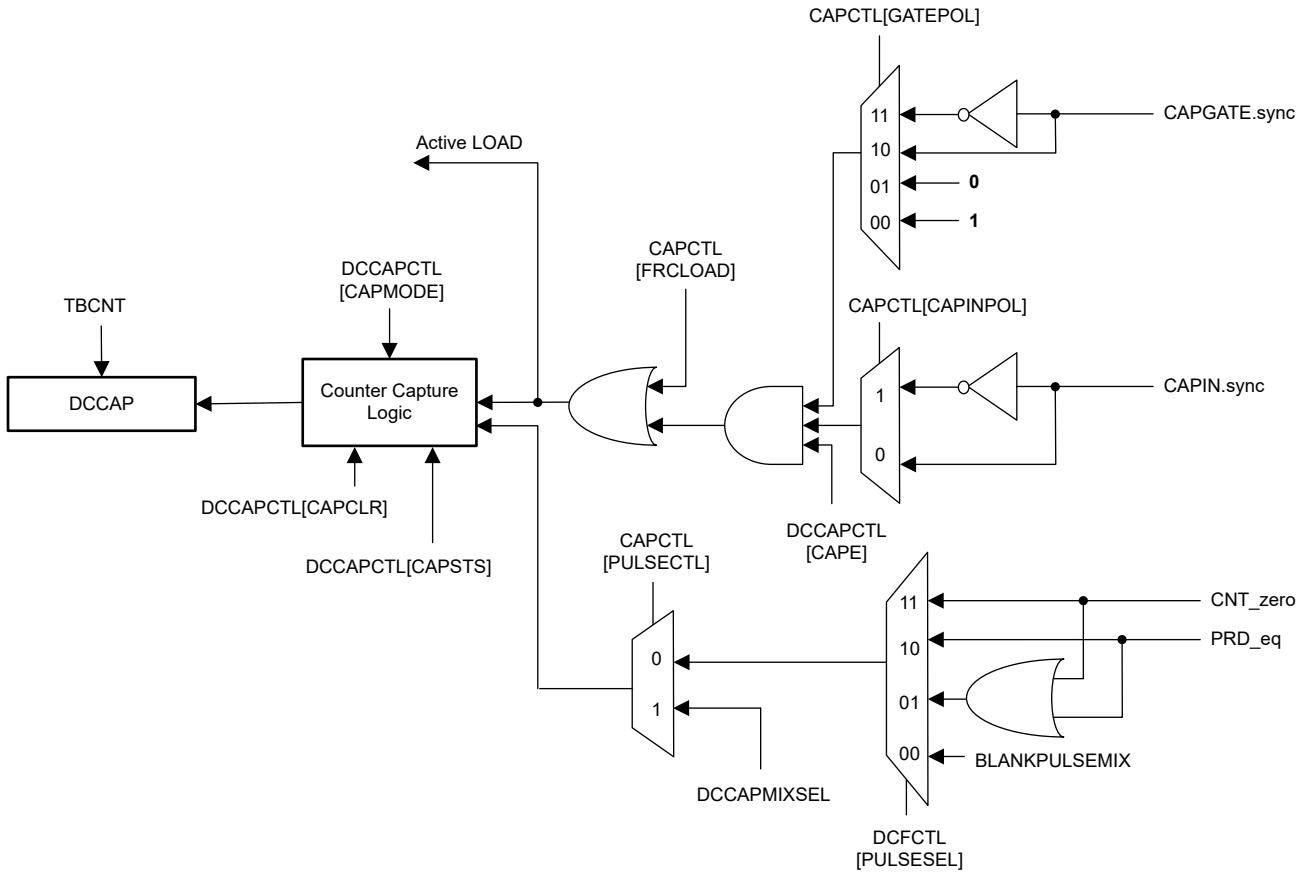


Figure 30-84. Counter Capture Logic

30.14.4.4.2 MIN and MAX Detection Circuit

The XMINMAX register has XMIN and XMAX fields that can be programmed to set the MIN and MAX limits of the programmable edge detection window. These registers have 3-level buffering similar to the XCMPn registers. The shadow to active loading of these registers is always in sync with the buffer pointers. Any shadow to active loads occur as per the XLOAD register configuration defined for the XCMPn registers such that the MIN and MAX values used are always in line with the corresponding XPRD/XCMPn values used for a given PWM cycle.

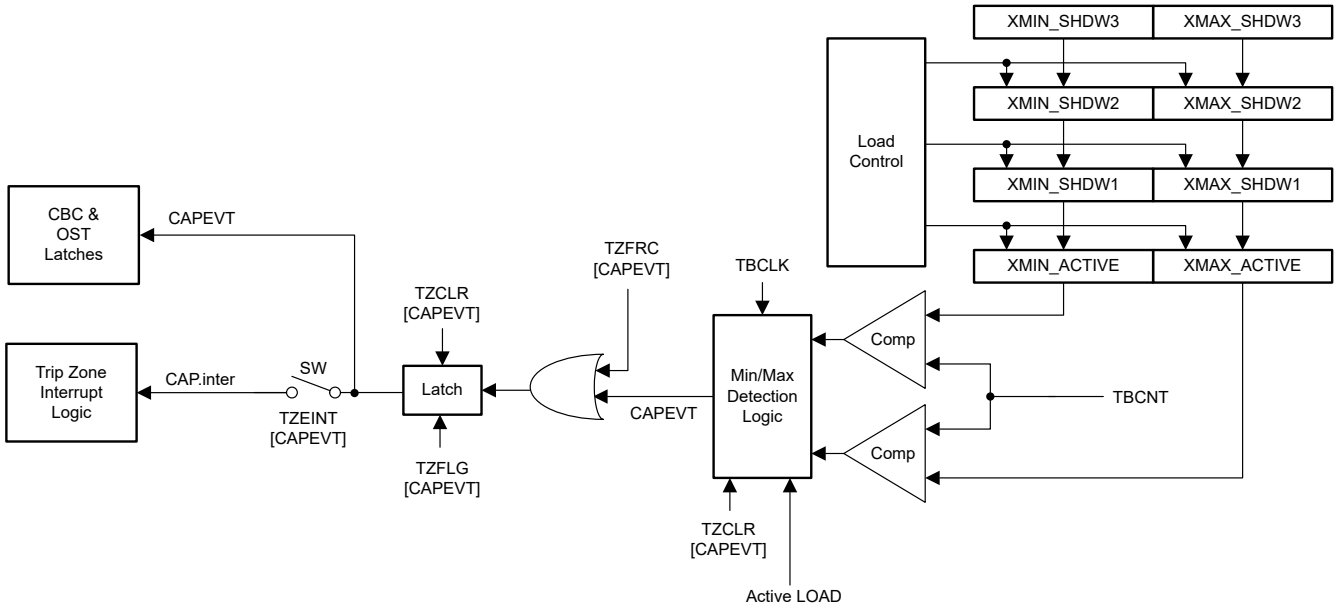


Figure 30-85. MIN and MAX Threshold Detection Logic

The logic works in the following way:

- The TBCNT value is continually monitored and compared against the active MIN value. Match of TBCNT to the active MIN value triggers the edge monitoring occurrence.
- When the TBCNT value reached the MIN value, the active LOAD signal is monitored waiting for an edge event to occur.
- If an edge vent occurs before TBCNT reaches the active MAX value, then no further action is taken. The logic resets and TBCNT is compared to the active MIN value again.
- If no edge occurs and TBCNT reaches the active MAX value, then the CAPEVT signal is set high and a CAP interrupt signal can also be generated. The CAPEVT signal needs to be cleared through software for TBCNT to be monitored against the MIN value again.

The Min and Max monitoring is enabled and disabled in three ways:

- By enabling/disabling the circuit via the DCCAPCTL[CAPE] bit
- By the CAPGATE signal which can be sourced from an TRIPINPUT signal to the module.
- By writing the same value into the XMIN and XMAX bits.

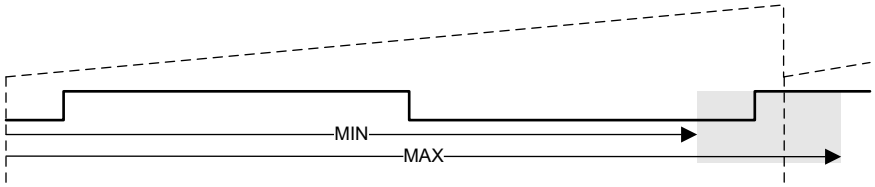


Figure 30-86. Capture Logic Boundary Condition

Note

Possible boundary condition of MIN/MAX window exceeding the period value: In this case, the XMAX bit can have a value lower than the XMIN bit such that the window can go over the period boundary.

30.15 ePWM Crossbar (X-BAR)

Figure 30-87 shows the architecture of the ePWM Crossbar (X-BAR). This module enables selection of various trigger sources into any of the dedicated EPWM trips inputs.

Note

Refer to the *Crossbar (X-BAR)* chapter for more information on the X-BAR modules, including X-BAR flags.

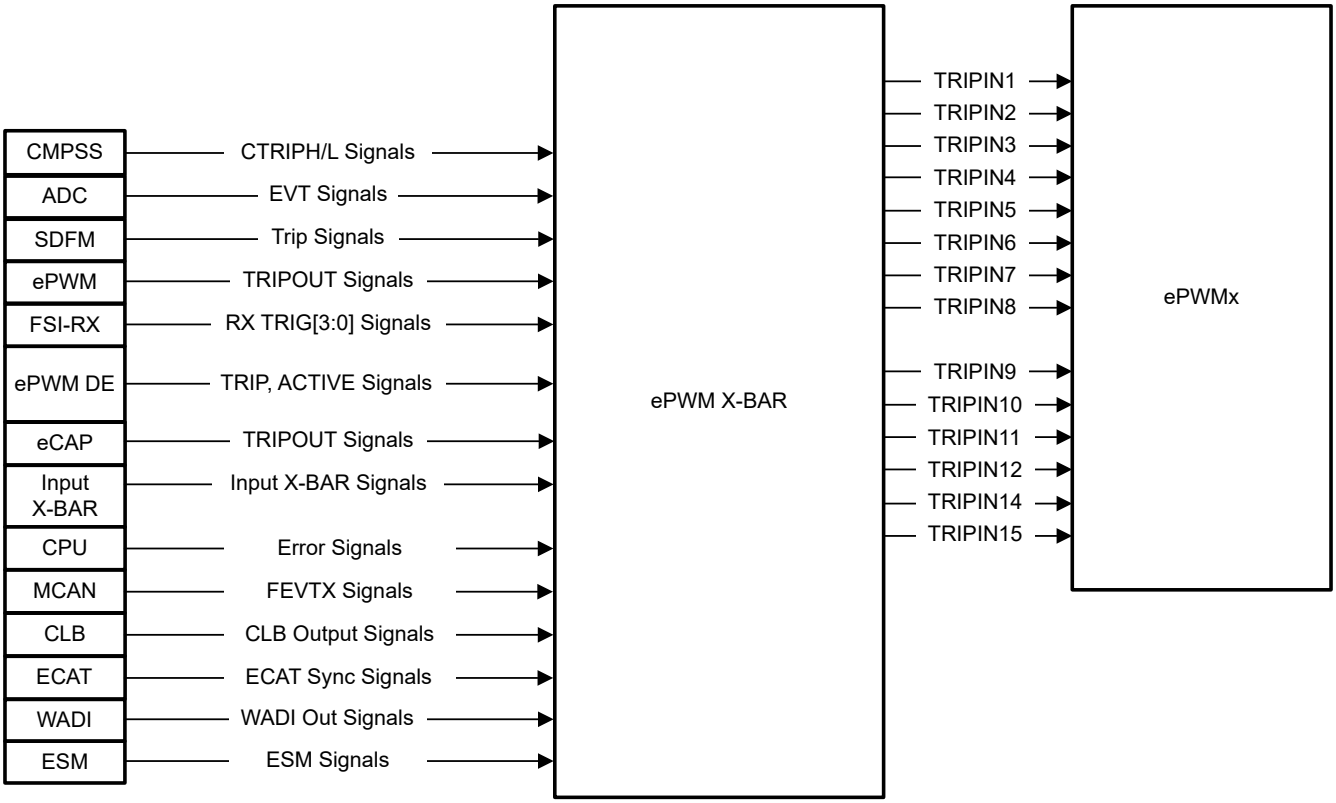


Figure 30-87. ePWM X-BAR

30.16 Applications to Power Topologies

An ePWM module has all the local resources necessary to operate completely as a standalone module or to operate in synchronization with other identical ePWM modules.

30.16.1 Overview of Multiple Modules

Previously in this chapter, all discussions have described the operation of a single module. To facilitate the understanding of multiple modules working together in a system, the ePWM module described in reference is represented by the more simplified block diagram shown in Figure 30-88. This simplified ePWM block shows only the key resources needed to explain how a multiswitch power topology is controlled with multiple ePWM modules working together.

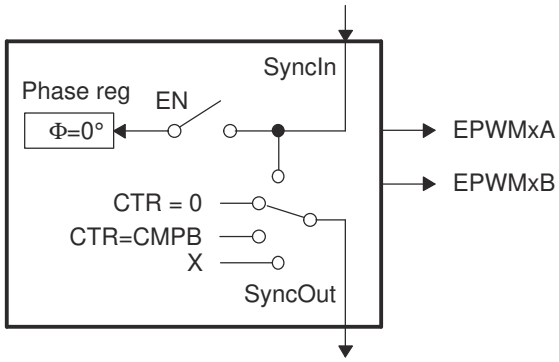


Figure 30-88. Simplified ePWM Module

30.16.2 Key Configuration Capabilities

The key configuration choices available to each module are as follows:

- Options for SyncIn
  - Load own counter with phase register on an incoming sync strobe—enable (EN) switch closed
  - Do nothing or ignore incoming sync strobe—enable switch open
  - Sync flow-through - SyncOut connected to SyncIn
  - Sync Source mode, provides a sync at PWM boundaries—SyncOut connected to CTR = PRD
  - Sync Source mode, provides a sync at any programmable point in time—SyncOut connected to CTR = CMPB
  - Module is in standalone mode and provides no sync to other modules—SyncOut connected to X (disabled)
- Options for SyncOut
  - Sync flow-through - SyncOut connected to SyncIn
  - Sync Source mode, provides a sync at PWM boundaries—SyncOut connected to CTR = PRD
  - Sync Source mode, provides a sync at any programmable point in time—SyncOut connected to CTR = CMPB
  - Module is in standalone mode and provides no sync to other modules—SyncOut connected to X (disabled)

For each choice of SyncOut, a module can also choose to load the counter with a new phase value on a SyncIn strobe input or choose to ignore the value (that is, by the enable switch). Although various combinations are possible, the two most common—Sync Source module and Sync Receiver module modes—are shown in Figure 30-89.

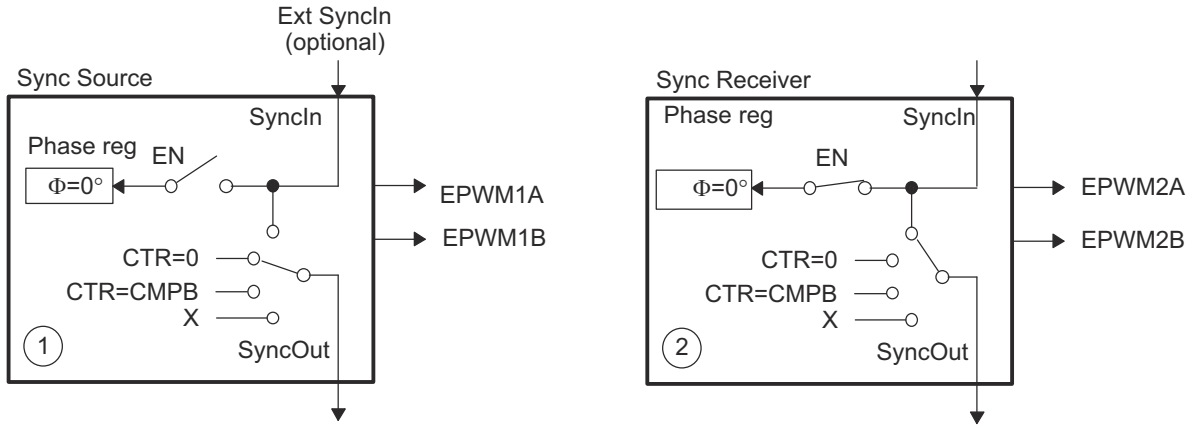
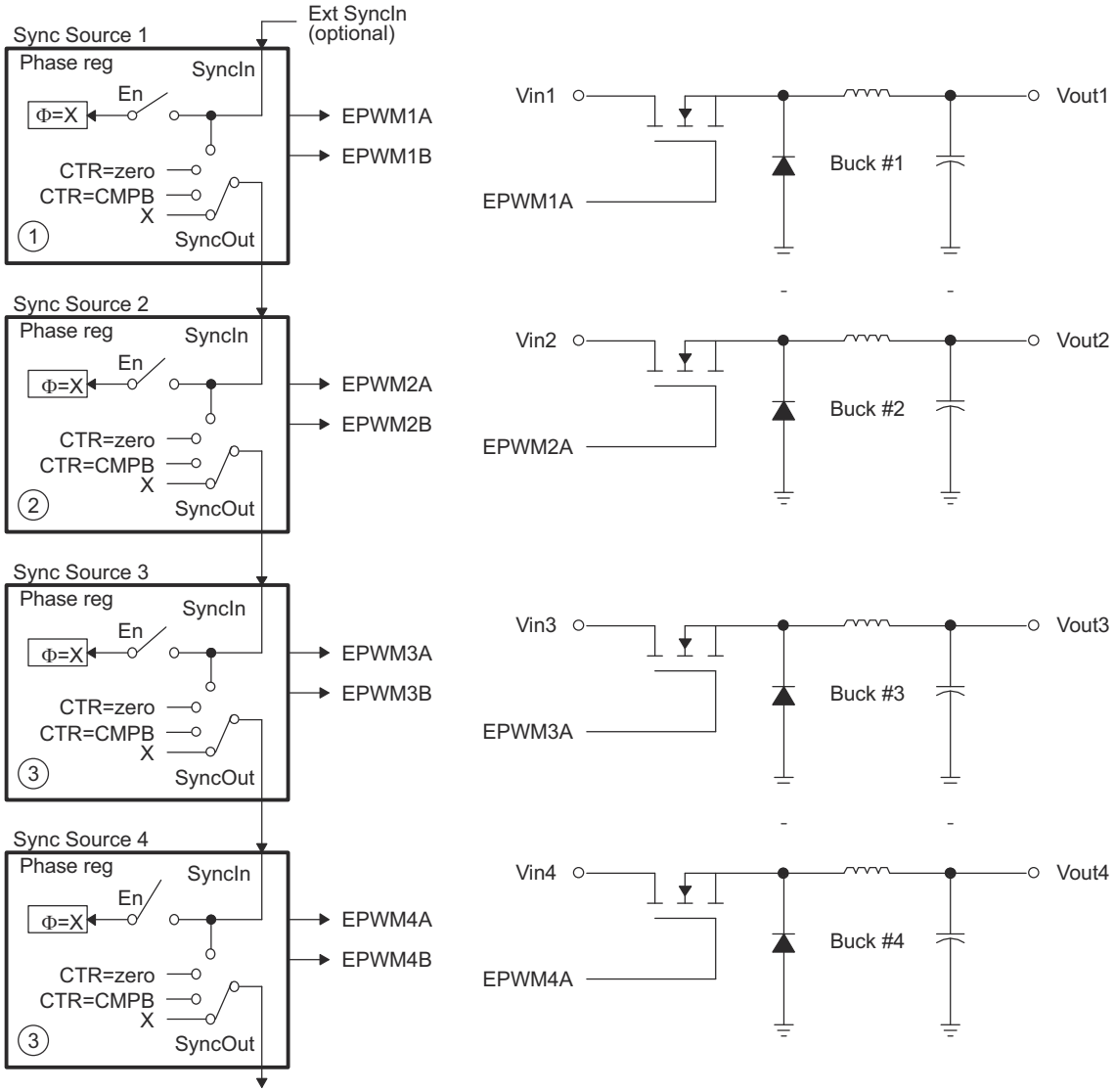


Figure 30-89. EPWM1 Configured as a Typical Sync Source, EPWM2 Configured as a Sync Receiver

30.16.3 Controlling Multiple Buck Converters With Independent Frequencies

One of the simplest power converter topologies is the buck. A single ePWM module configured as a sync source can control two buck stages with the same PWM frequency. If independent frequency control is required for each buck converter, then one ePWM module must be allocated for each converter stage. Figure 30-90 shows four buck stages, each running at independent frequencies. In this case, all four ePWM modules are configured as Sync Sources and no synchronization is used. Figure 30-91 shows the waveforms generated by the setup shown in Figure 30-90; note that only three waveforms are shown, although there are four stages.



A. φ = X indicates value in phase register is a "don't care"

Figure 30-90. Control of Four Buck Stages. Here $F_{PWM1} \neq F_{PWM2} \neq F_{PWM3} \neq F_{PWM4}$

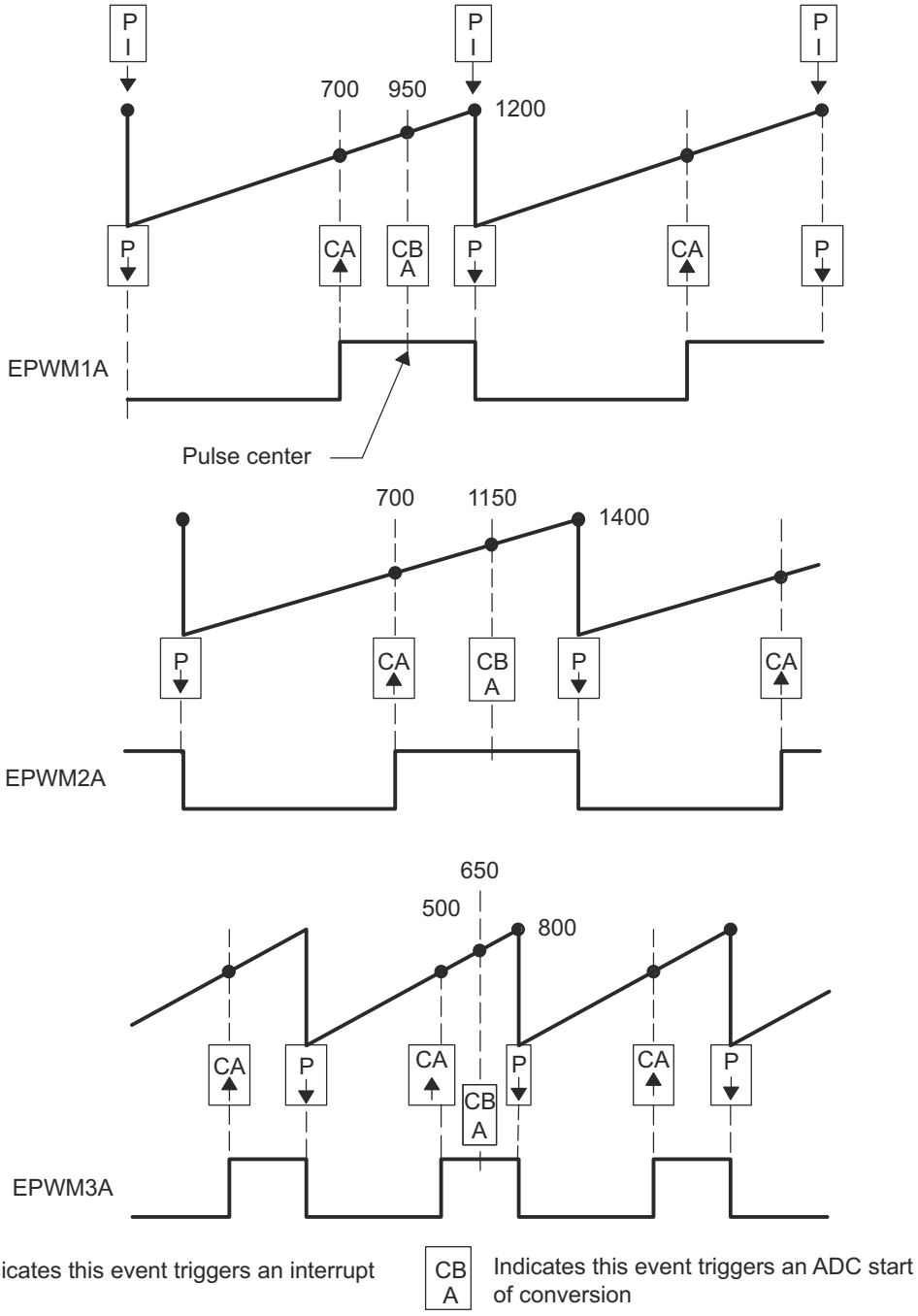
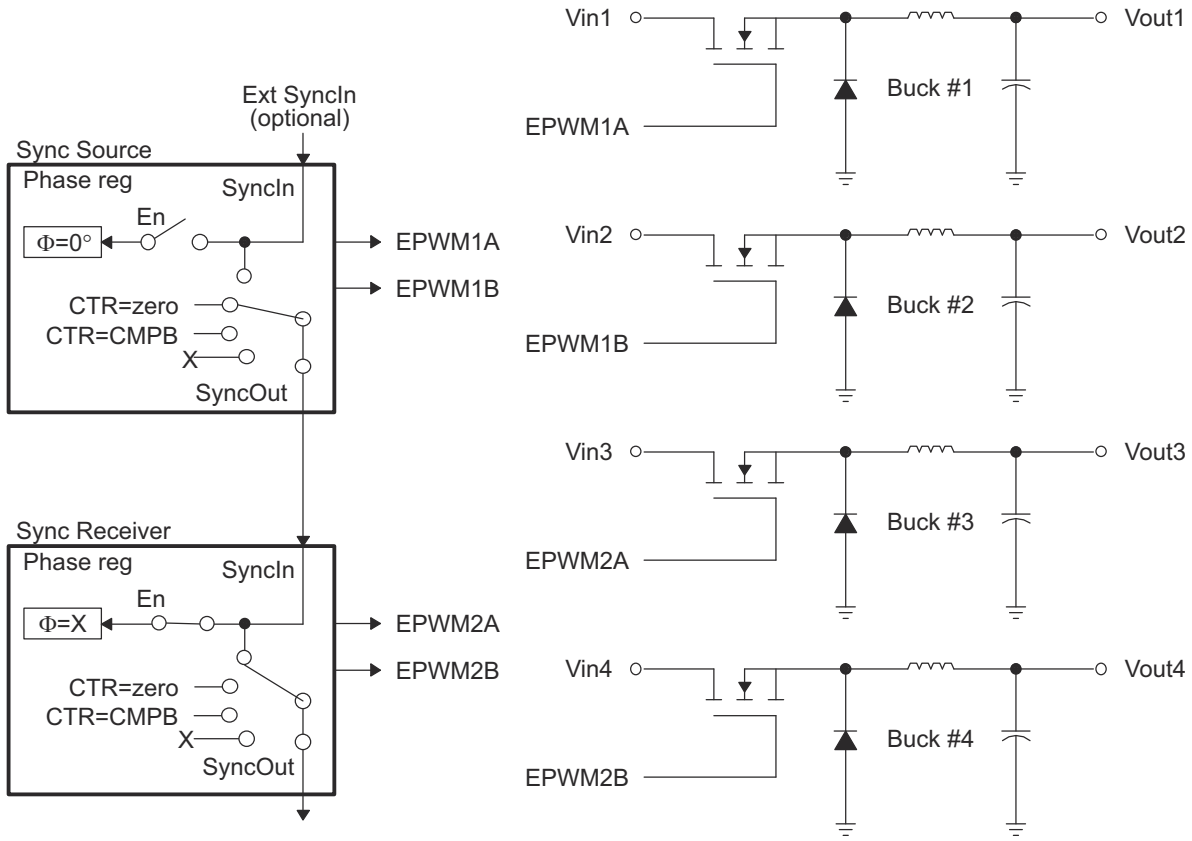


Figure 30-91. Buck Waveforms for Control of Four Buck Stages (Note: Only three bucks shown here)

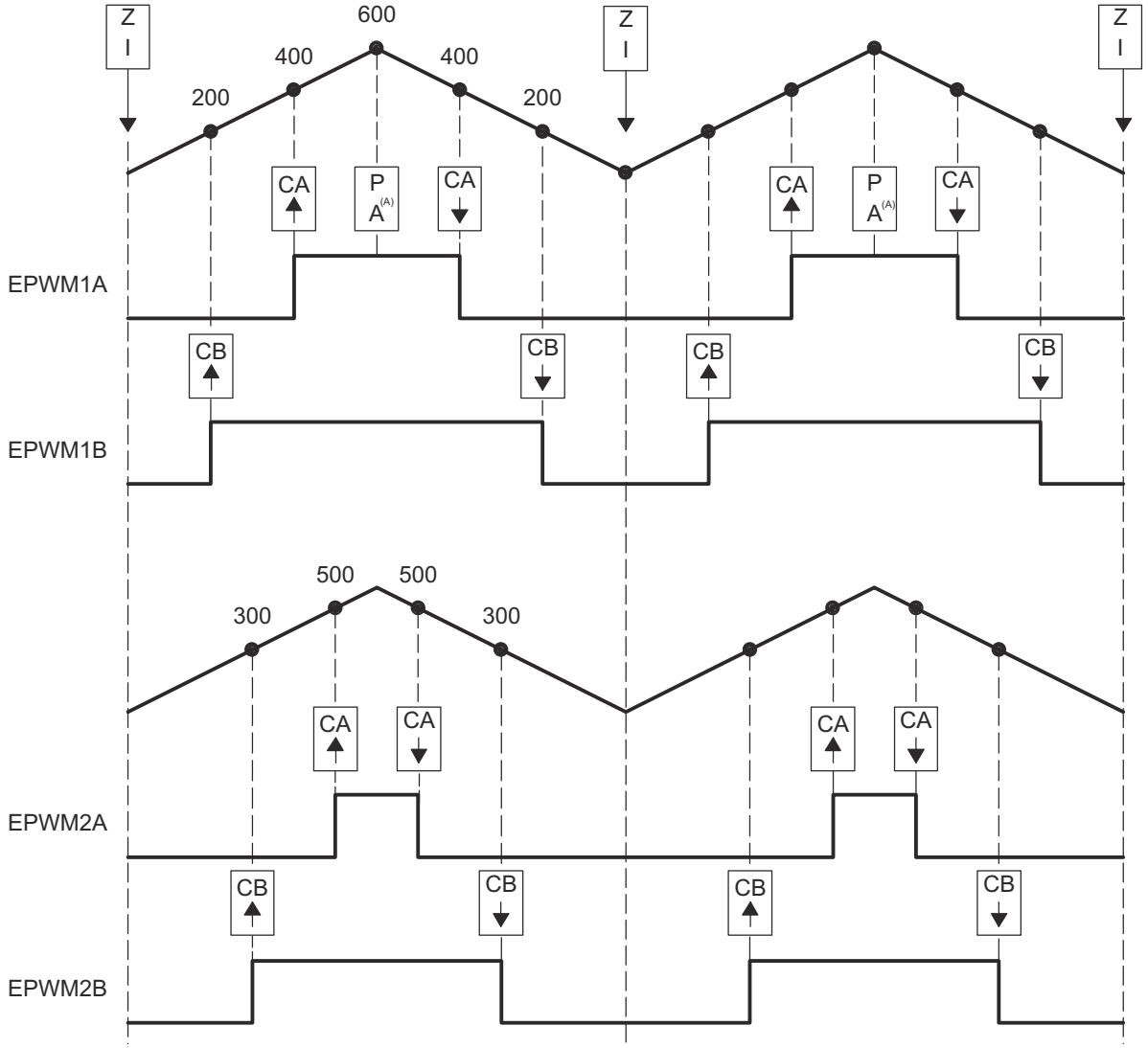
30.16.4 Controlling Multiple Buck Converters With Same Frequencies

If synchronization is a requirement, ePWM module 2 is configured as a sync receiver and operates at integer multiple (N) frequencies of module 1. The sync signal from sync source to sync receiver makes sure these modules remain locked. Figure 30-92 shows such a configuration; Figure 30-93 shows the waveforms generated by the configuration.



A. φ = X indicates value in phase register is a "don't care"

Figure 30-92. Control of Four Buck Stages. (Note: F_{PWM2} = N x F_{PWM1})



A. Starts ADC conversion.

Figure 30-93. Buck Waveforms for Control of Four Buck Stages (Note: $F_{PWM2} = F_{PWM1}$)

30.16.5 Controlling Multiple Half H-Bridge (HHB) Converters

Topologies that require control of multiple switching elements can also be addressed with these same ePWM modules. It is possible to control a Half-H bridge stage with a single ePWM module. This control can be extended to multiple stages. Figure 30-94 shows control of two synchronized Half-H bridge stages where stage 2 can operate at integer multiple (N) frequencies of stage 1. Figure 30-95 shows the waveforms generated by the configuration shown in Figure 30-94.

ePWM module 2 (sync receiver) is configured for Sync flow-through; if required, this configuration allows for a third Half-H bridge to be controlled by ePWM module 3 and also, most importantly, to remain in synchronization with sync source ePWM module 1.

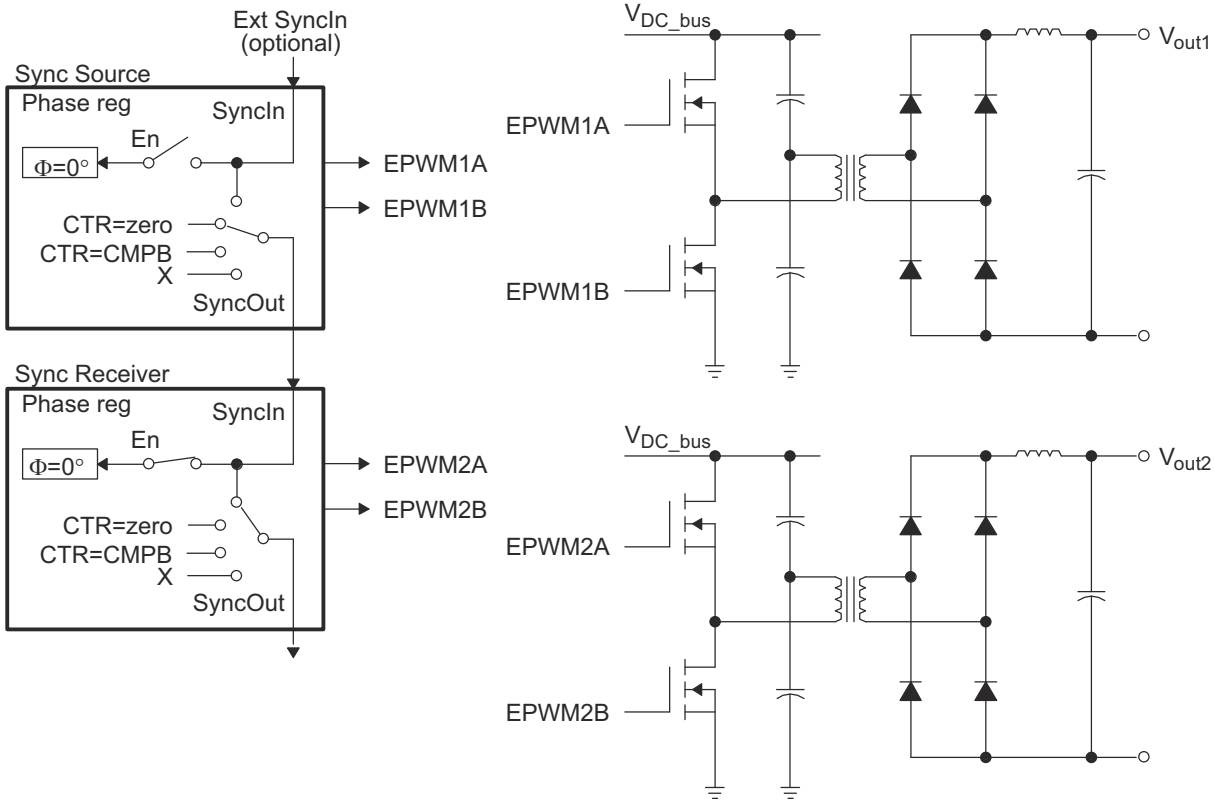


Figure 30-94. Control of Two Half-H Bridge Stages (F_{PWM2} = N x F_{PWM1})

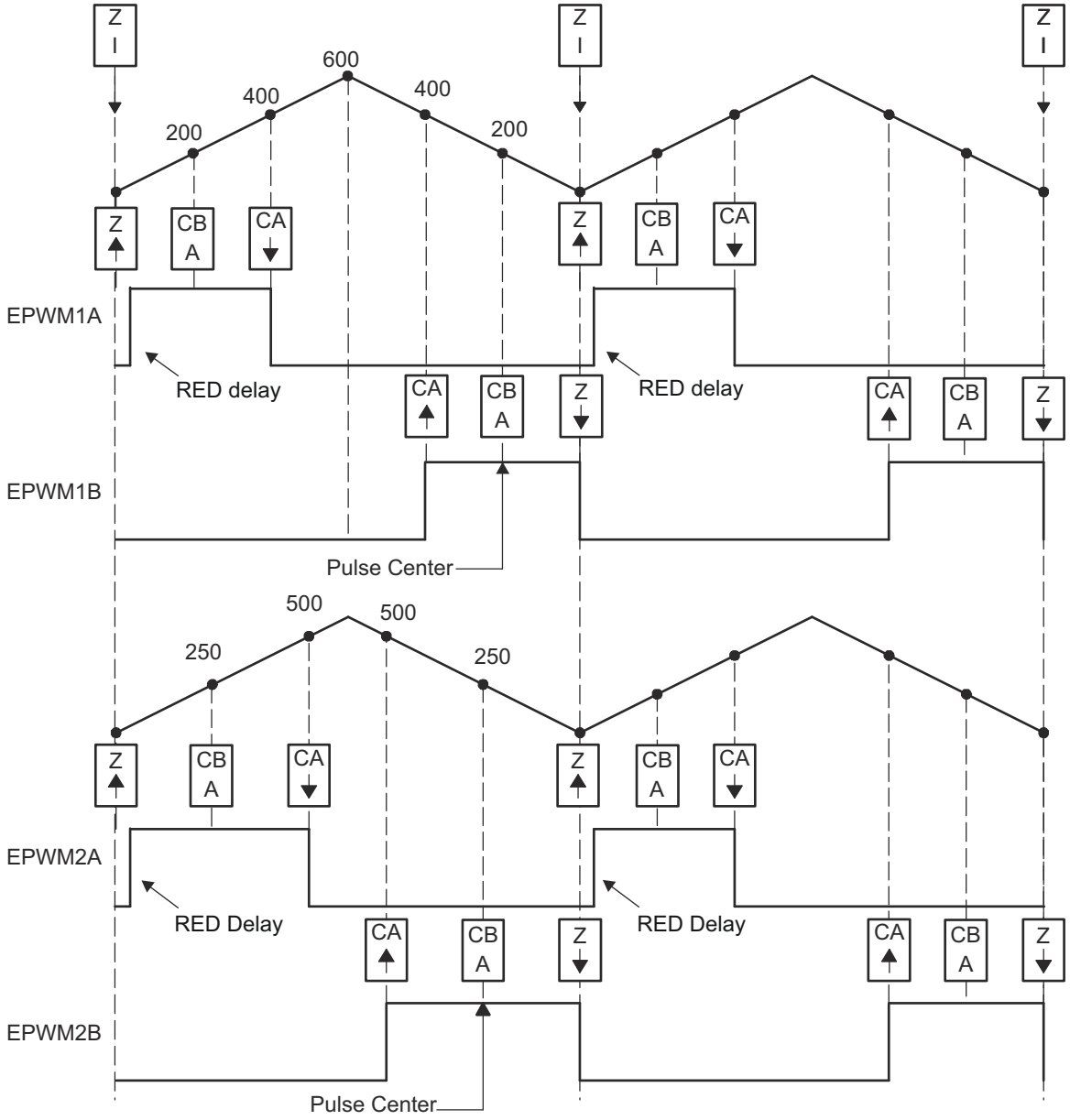


Figure 30-95. Half-H Bridge Waveforms for Control of Two Half-H Bridge Stages (Note: Here $F_{PWM2} = F_{PWM1}$)

30.16.6 Controlling Dual 3-Phase Inverters for Motors (ACI and PMSM)

The idea of multiple modules controlling a single power stage can be extended to the 3-phase inverter case. In such a case, six switching elements are controlled using three PWM modules, one for each leg of the inverter. Each leg must switch at the same frequency and all legs must be synchronized. A sync receivers configuration easily addresses this requirement. Figure 30-96 shows how six PWM modules control two independent 3-phase inverters; each running a motor.

As in the cases shown in the previous sections, we have a choice of running each inverter at a different frequency (module 1 and module 4 are sync sources as in Figure 30-96), or both inverters can be synchronized by using one sync source (module 1) and five sync receivers. In this case, the frequency of modules 4, 5, and 6 (all equal) can be integer multiples of the frequency for modules 1, 2, and 3 (also all equal).

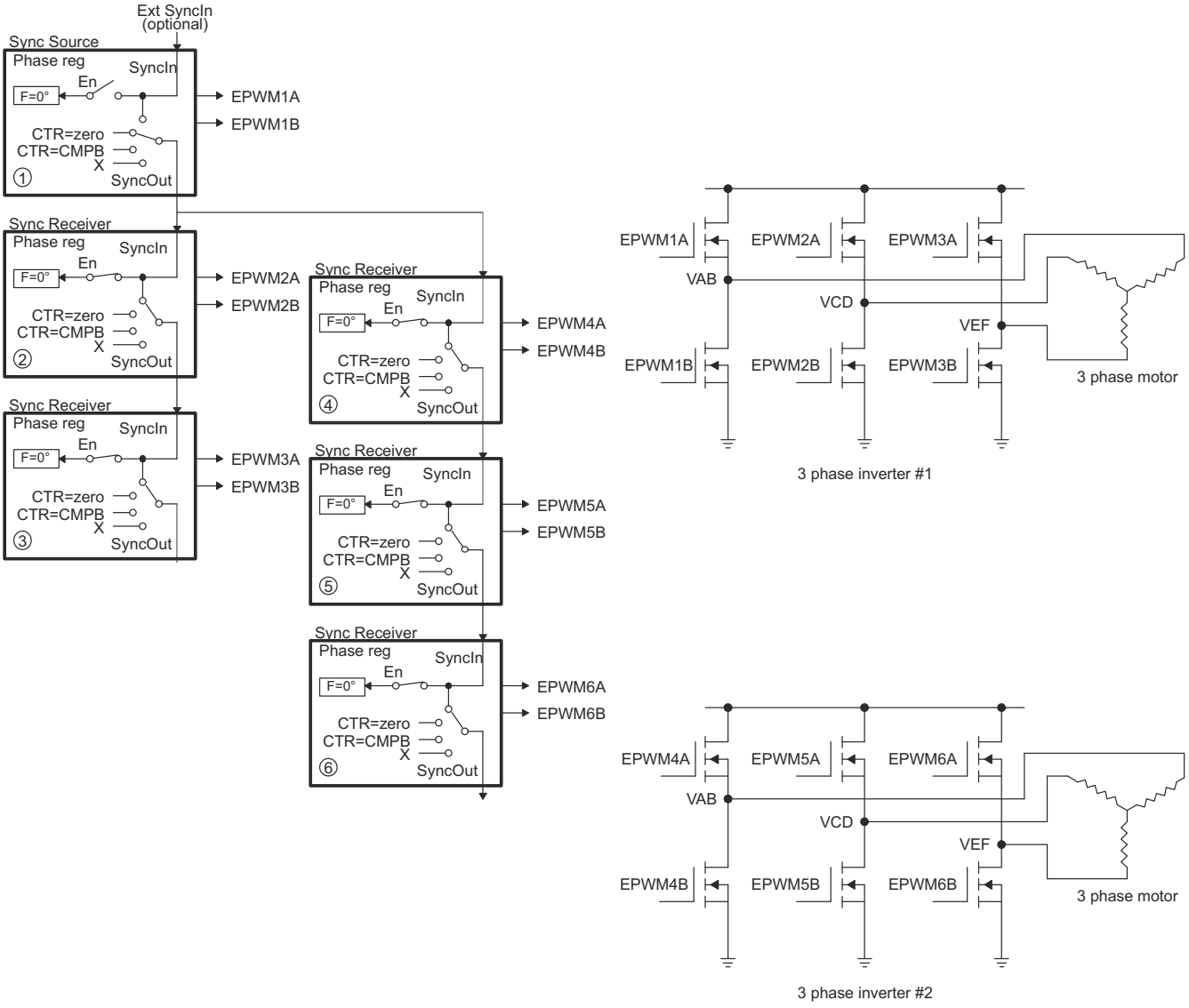


Figure 30-96. Control of Dual 3-Phase Inverter Stages as Is Commonly Used in Motor Control

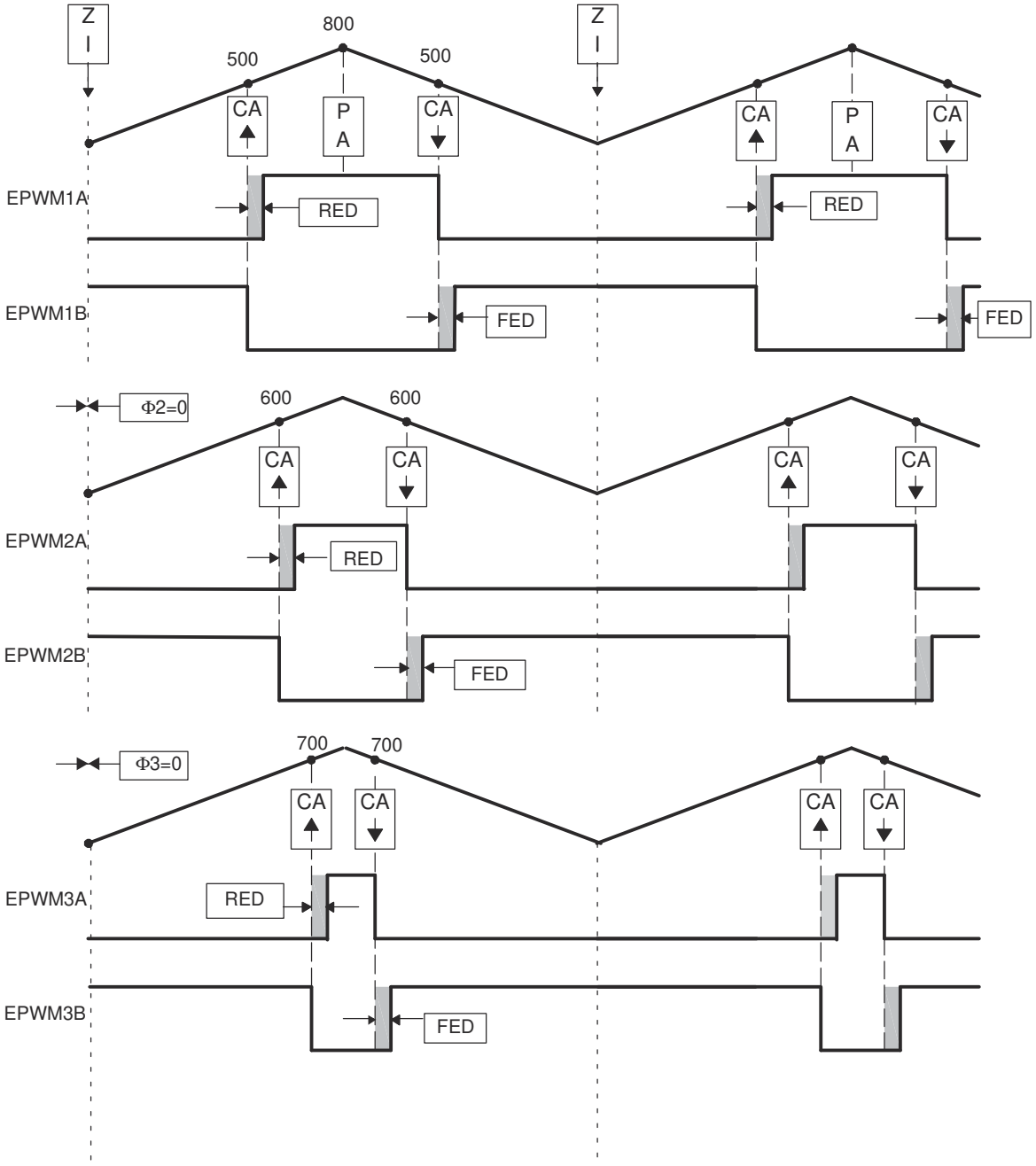


Figure 30-97. 3-Phase Inverter Waveforms for Control of Dual 3-Phase Inverter Stages (Only One Inverter Shown)

30.16.7 Practical Applications Using Phase Control Between PWM Modules

So far, none of the examples have made use of the phase register (TBPHS). It has either been set to zero or a don't care. However, by programming appropriate values into TBPHS, multiple PWM modules can address another class of power topologies that rely on phase relationship between legs (or stages) for correct operation. As described in the time-base submodule section, a PWM module can be configured to allow a SyncIn pulse to cause the TBPHS register to be loaded into the TBCTR register. To illustrate this concept, Figure 30-98 shows a sync source and sync receiver module with a phase relationship of 120° (that is, the sync receiver leads the sync source).

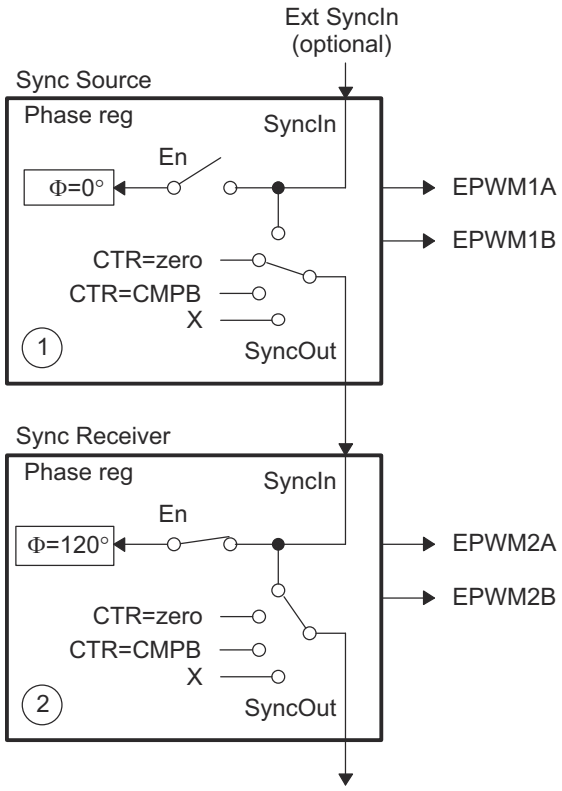


Figure 30-98. Configuring Two PWM Modules for Phase Control

Figure 30-99 shows the associated timing waveforms for this configuration. Here, TBPRD = 600 for both sync source and sync receiver. For the sync receiver, TBPHS = 200 (that is, 200/600 x 360° = 120°). Whenever the sync source generates a SyncIn pulse (CTR = PRD), the value of TBPHS = 200 is loaded into the sync receiver TBCTR register so the sync receiver time-base is always leading the sync source time-base by 120°.

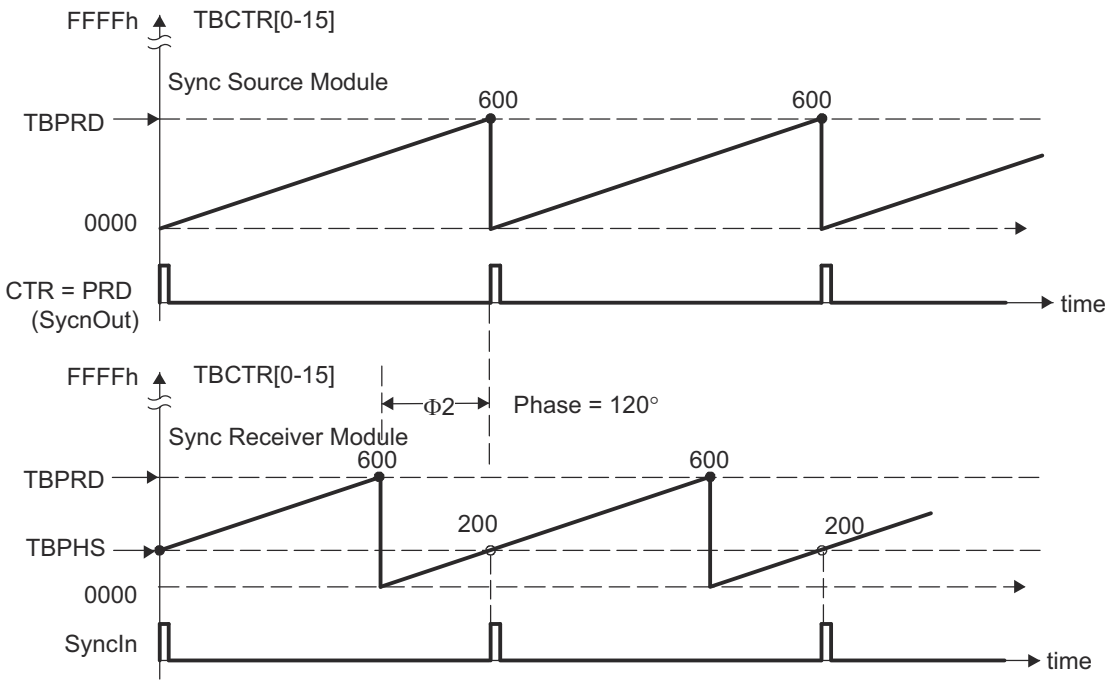


Figure 30-99. Timing Waveforms Associated with Phase Control Between Two Modules

30.16.8 Controlling a 3-Phase Interleaved DC/DC Converter

A popular power topology that makes use of phase-offset between modules is shown in Figure 30-100. This system uses three PWM modules, with module 1 configured as the sync source. To work, the phase relationship between adjacent modules must be F = 120°. This is achieved by setting the sync receiver TBPHS registers 2 and 3 with values of 1/3 and 2/3 of the period value, respectively. For example, if the period register is loaded with a value of 600 counts, then TBPHS (sync receiver 2) = 200 and TBPHS (sync receiver 3) = 400. Both sync receiver modules are synchronized to the sync source module 1.

This concept can be extended to four or more phases, by setting the TBPHS values appropriately. The following formula gives the TBPHS values for N phases:

$$TBPHS(N,M) = (TBPRD/N) \times (M - 1)$$

Where:

N = number of phases

M = PWM module number

For example, for the 3-phase case (N = 3), TBPRD = 600,

TBPHS(3,2) = (600/3) × (2 - 1) = 200 (that is, Phase value for Sync Receiver module 2)

TBPHS(3,3) = 400 (that is, Phase value for Sync Receiver module 3)

Figure 30-101 shows the waveforms for the configuration in Figure 30-100.

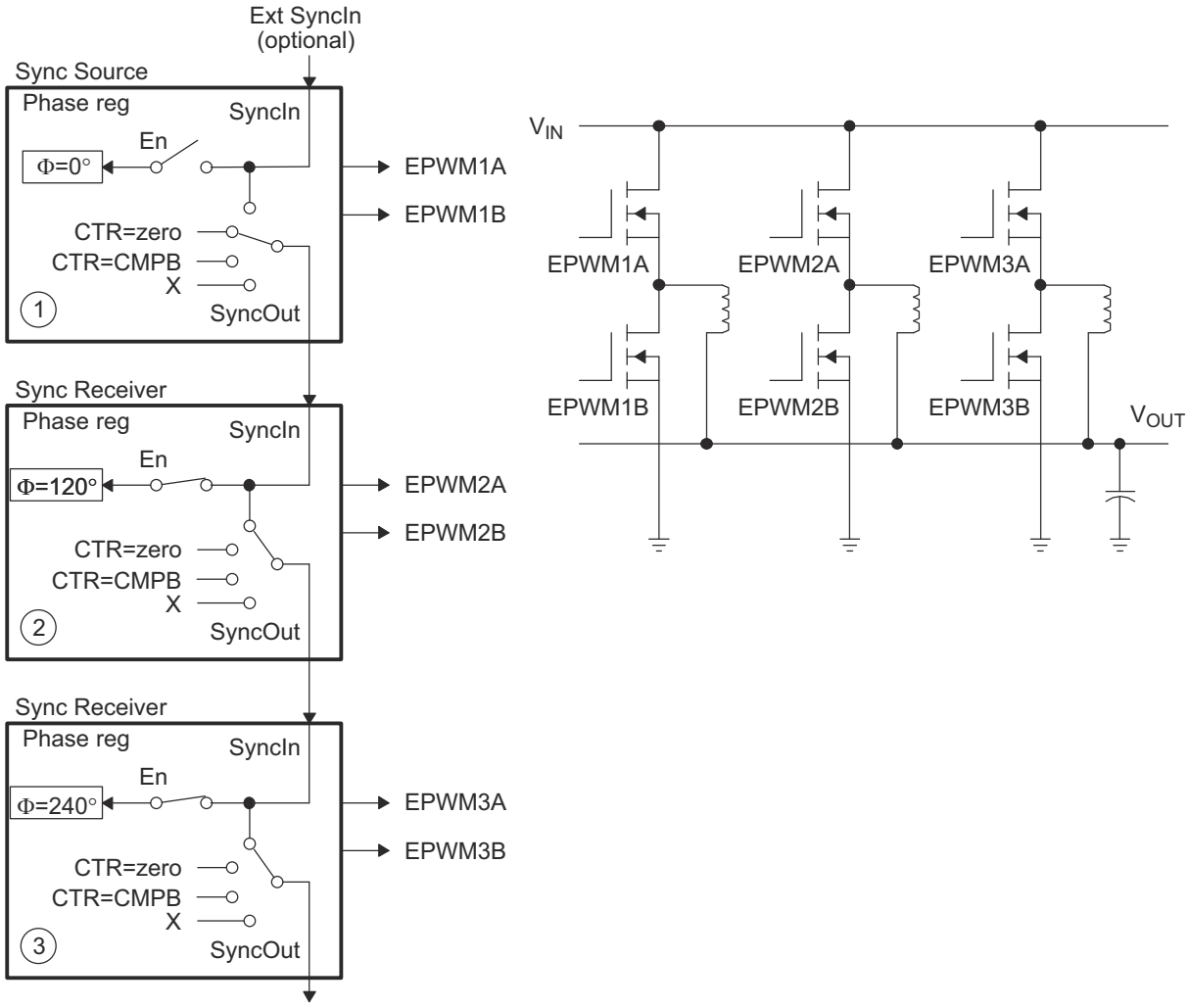


Figure 30-100. Control of 3-Phase Interleaved DC/DC Converter

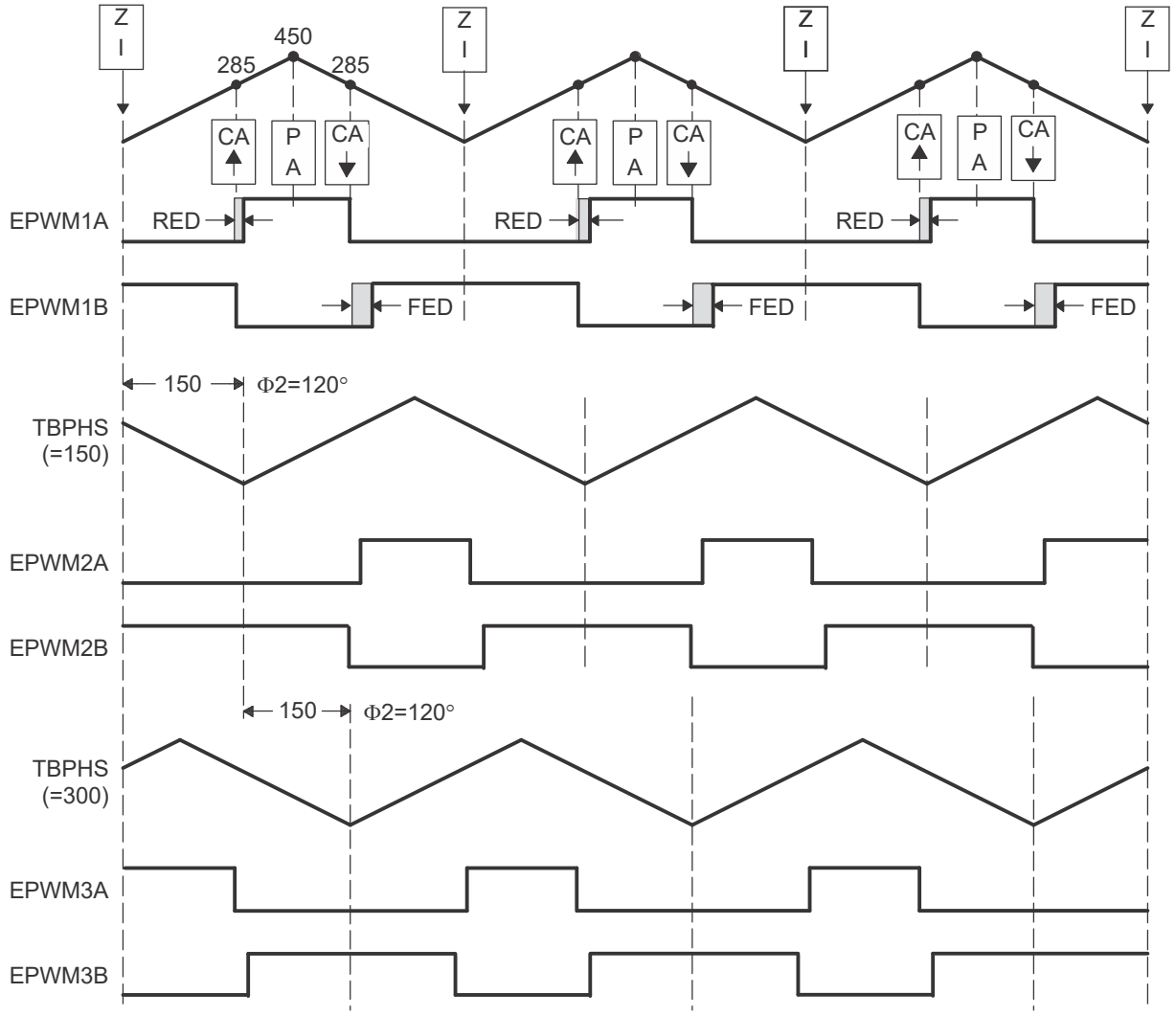


Figure 30-101. 3-Phase Interleaved DC/DC Converter Waveforms for Control of 3-Phase Interleaved DC/DC Converter

30.16.9 Controlling Zero Voltage Switched Full Bridge (ZVSFB) Converter

The example given in Figure 30-102 assumes a static or constant phase relationship between legs (modules). In such a case, control is achieved by modulating the duty cycle. It is also possible to dynamically change the phase value on a cycle-by-cycle basis. This feature lends to controlling a class of power topologies known as *phase-shifted full bridge*, or *zero voltage switched full bridge.* Here the controlled parameter is not duty cycle (this is kept constant at approximately 50 percent); instead it is the phase relationship between legs. Such a system can be implemented by allocating the resources of two PWM modules to control a single power stage, which in turn requires control of four switching elements. Figure 30-103 shows a sync source and sync receiver module combination synchronized together to control a full H-bridge. In this case, both sync source and sync receiver modules are required to switch at the same PWM frequency. The phase is controlled by using the sync receiver phase register (TBPHS). The sync source phase register is not used and therefore can be initialized to zero.

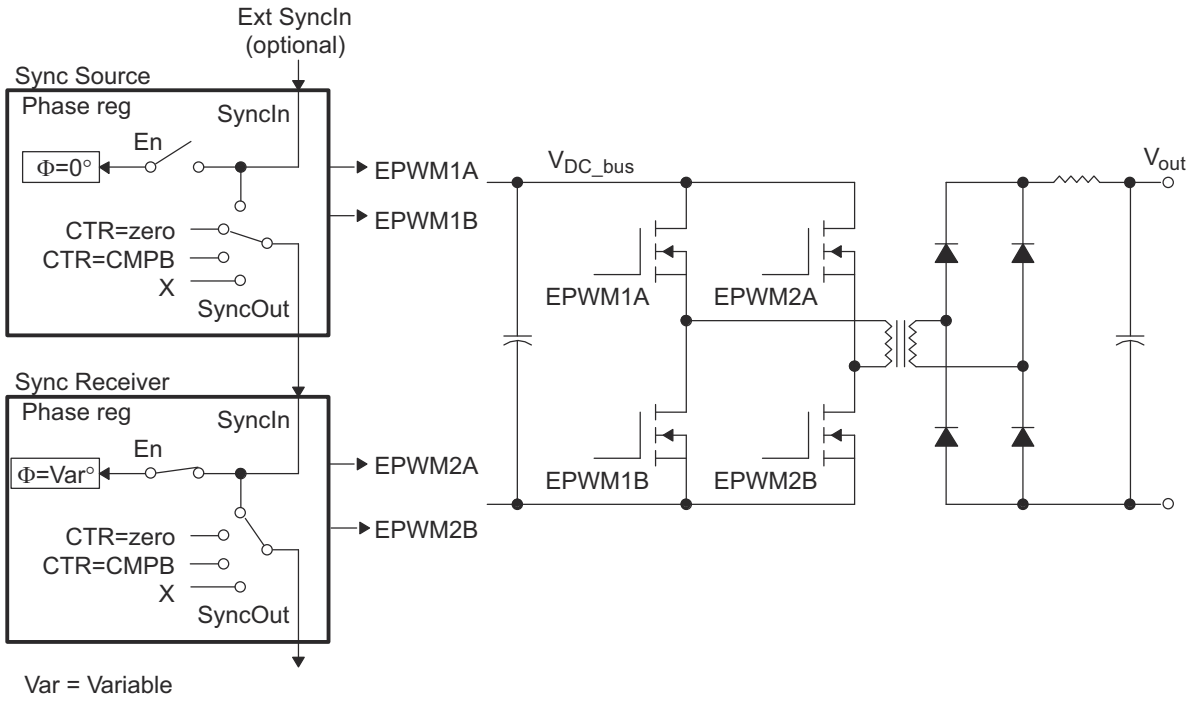


Figure 30-102. Control of Full-H Bridge Stage (F_{PWM2} = F_{PWM1})

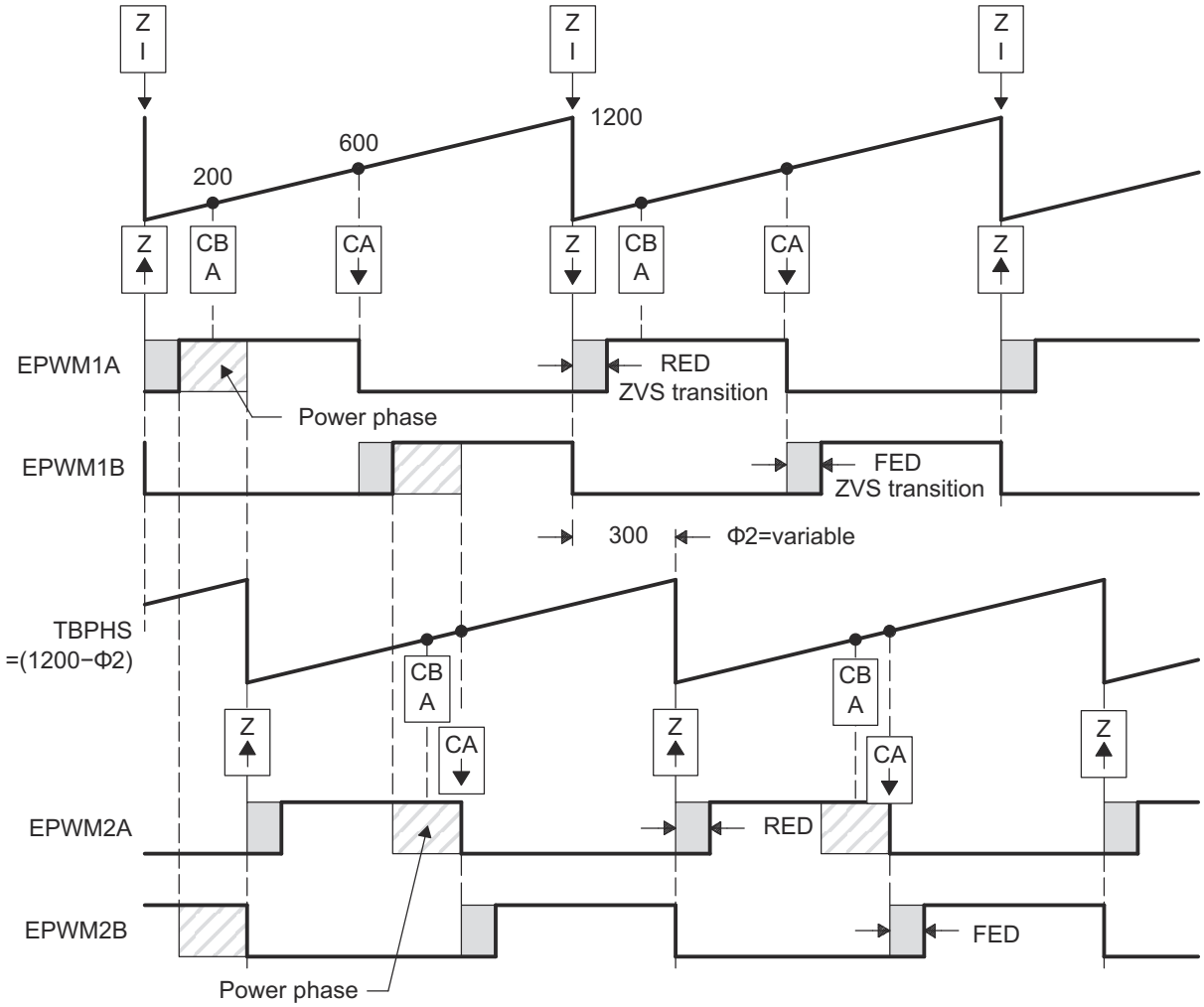


Figure 30-103. ZVS Full-H Bridge Waveforms

30.16.10 Controlling a Peak Current Mode Controlled Buck Module

Peak current control techniques offer a number of benefits like automatic over current limiting, fast correction for input voltage variations and reducing magnetic saturation. Figure 30-104 shows the use of ePWM1A along with the on-chip analog comparator for buck converter topology. The output current is sensed through a current sense resistor and fed to the positive terminal of the on-chip comparator. The internal programmable 12-bit DAC can be used to provide a reference peak current at the negative terminal of the comparator. Alternatively, an external reference can be connected at this input. The comparator output is an input to the Digital compare sub-module. The ePWM module is configured in such a way so as to trip the ePWM1A output as soon as the sensed current reaches the peak reference value. A cycle-by-cycle trip mechanism is used. Figure 30-105 shows the waveforms generated by the configuration.

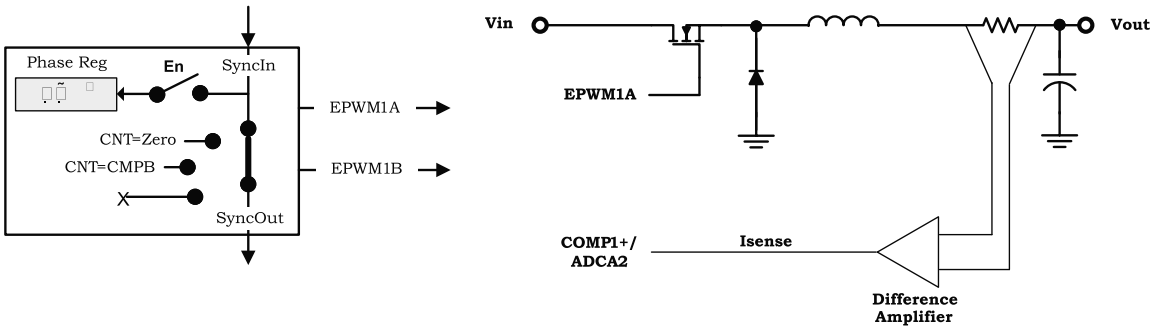


Figure 30-104. Peak Current Mode Control of Buck Converter

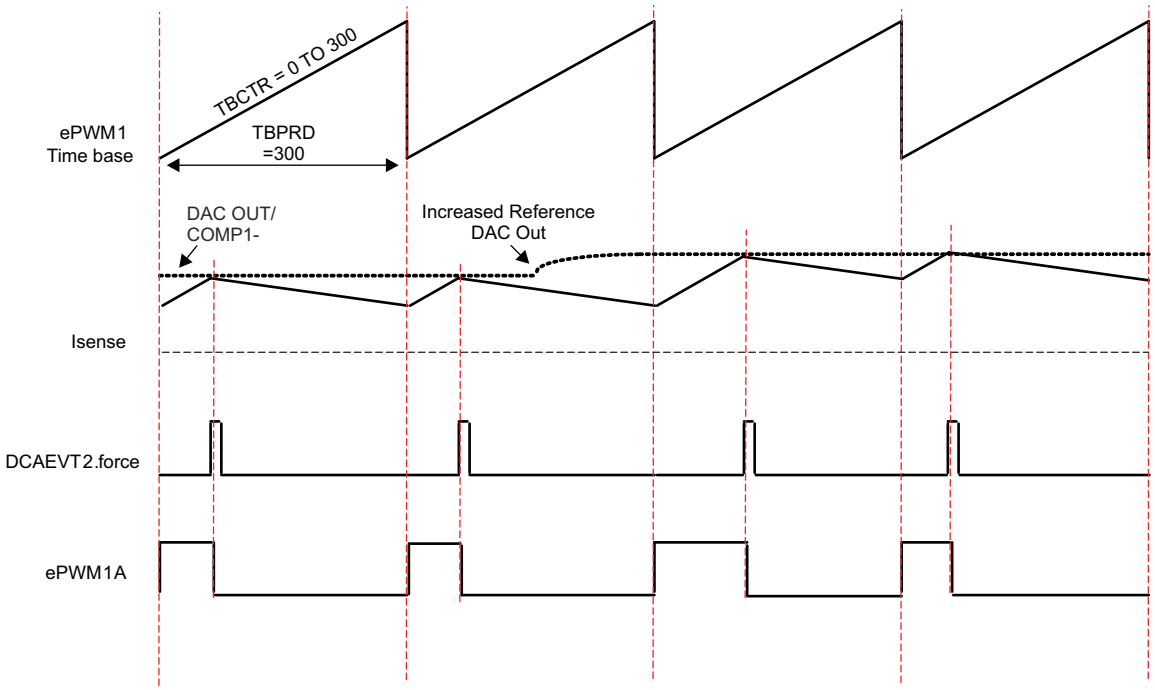
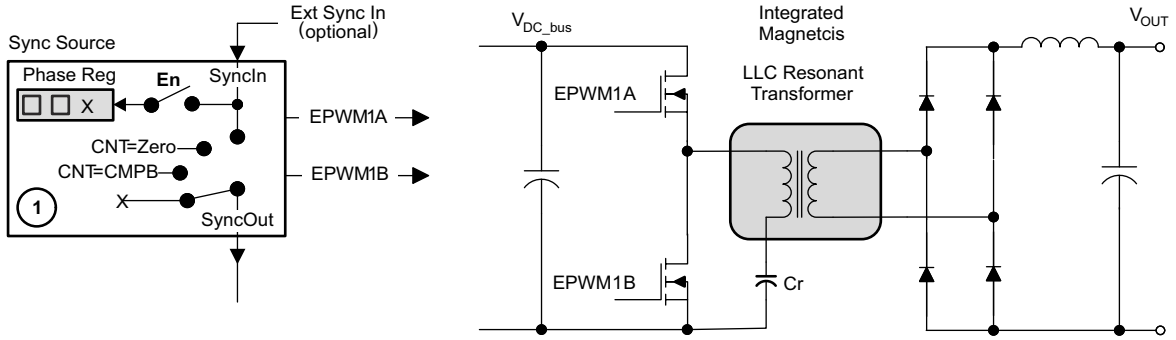


Figure 30-105. Peak Current Mode Control Waveforms for Control of Buck Converter

30.16.11 Controlling H-Bridge LLC Resonant Converter

Various topologies of resonant converters are well-known in the field of power electronics for many years. In addition to these, H-bridge LLC resonant converter topology has recently gained popularity in many consumer electronics applications where high efficiency and power density are required. In this example, single channel configuration of ePWM1 is detailed, yet the configuration can easily be extended to multichannel. Here the controlled parameter is not duty cycle (this is kept constant at approximately 50 percent); instead the parameter is frequency. Although the deadband is not controlled and kept constant as 300ns (that is, 30 at 100MHz TBCLK), the user can update the deadband in real time to enhance the efficiency by adjusting enough time delay for soft switching.



NOTE Θ = X indicates value in phase register is a “don’t care”

Figure 30-106. Control of Two Resonant Converter Stages

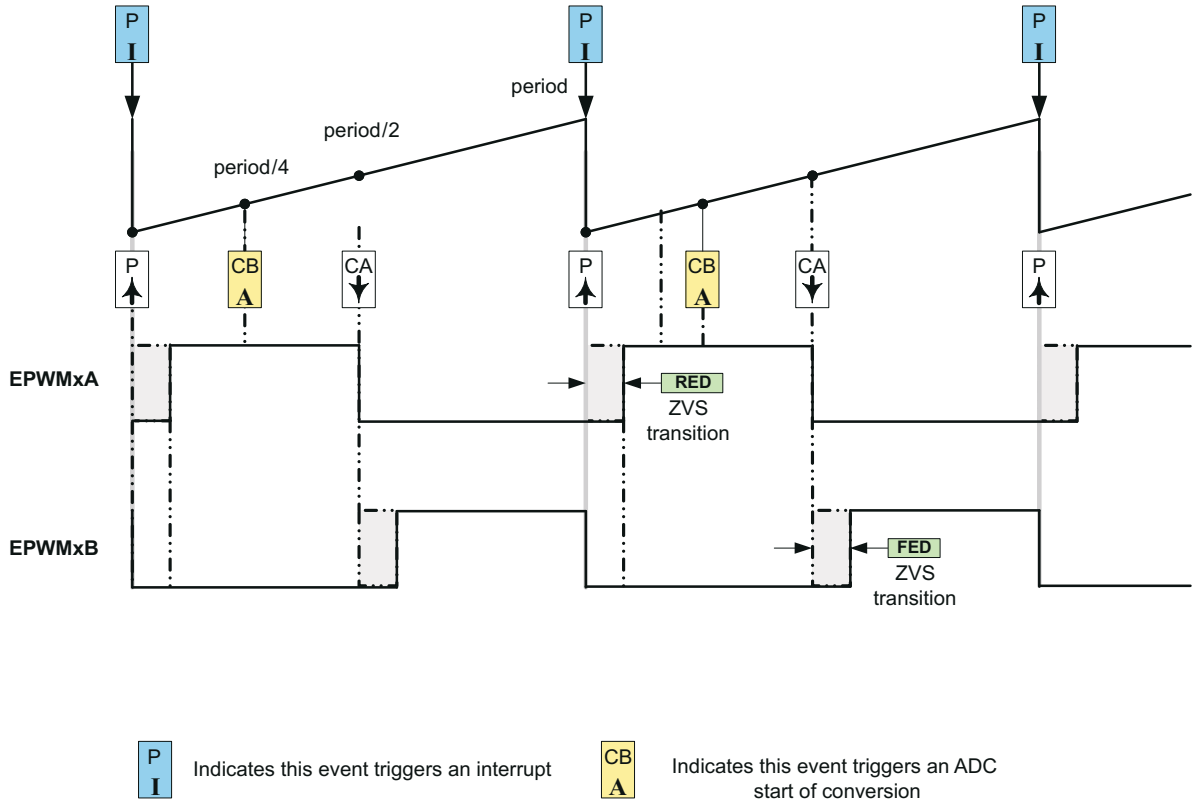


Figure 30-107. H-Bridge LLC Resonant Converter PWM Waveforms

30.17 Register Lock Protection

The register lock protection mechanism is added to protect the critical ePWM registers from being corrupted by accidental writes in case of runaway code. The register EPWMLOCK contains the definition of Lock bits (Table 30-15 shows the lock bits and the corresponding registers). This register also has a KEY field; writes to this register succeed only if the KEY field is written with a value of 0xa5a5. Refer to the register descriptions for more details.

Table 30-15. Lock Bits and Corresponding Registers

Bit Field	Definition	Registers Locked
HRLOCK	HRPWM Register Set Lock	HRCNFG, HRPWR, HRMSTEP, HRPCTL
GLLOCK	Global Load Register Set Lock	GLDCTL, GLDCFG
TZCFGLOCK	TripZone Register Set Lock	TZSEL, TZDCSEL, TZCTL, TZCTL2, TZCTLDCA, TZCTLDCB, TZEINT
TZCLRLOCK	TripZone Clear Register Set Lock	TZCLR, TZCBCCLR, TZOSTCLR, TZFRC
DCLOCK	Digital Compare Register Set Lock	DCTRIPSEL, DCACTL, DCBCTL, DCFCTL, DCCAPCTL, DCAHTRIPSEL, DCALTRIPSEL, DCBHTRIPSEL, DCBLTRIPSEL

Note

Due to the presence of the KEY field in the same register, only 32-bit writes succeed if the KEY matches. The 16-bit writes to the upper or lower half of this register are ignored.

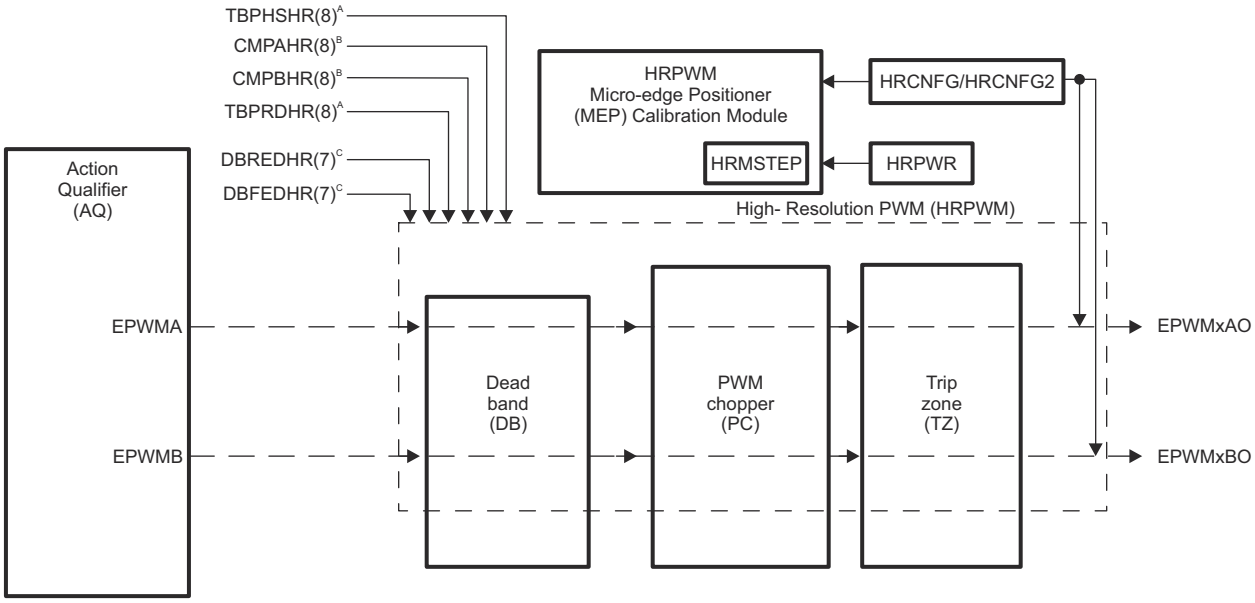
30.18 High-Resolution Pulse Width Modulator (HRPWM)

Figure 30-108 shows a block diagram of the HRPWM. This module extends the time resolution capabilities of the conventionally derived digital pulse width modulator (PWM). HRPWM is typically used when PWM resolution falls below approximately 9-10 bits. The key features of HRPWM are:

- Extended time resolution capability
- Used in both duty cycle and phase-shift control methods
- Finer time granularity control or edge positioning using extensions to the Compare A, Compare B and Phase registers
- Implemented using the A and B signal path of PWM, that is, on the EPWMxA and EPWMxB output
- Dead band high-resolution control for falling and rising edge delay in half cycle clocking operation
- Self-check diagnostics software mode to check if the micro edge positioner (MEP) logic is running how designed
- Enables high-resolution output swapping on the EPWMxA and EPWMxB output
- Enables high-resolution output on EPWMxB signal output using inversion of EPWMxA signal output
- Enables high-resolution period, duty and phase control on the EPWMxA and EPWMxB output on devices with an ePWM module

Note

See the device data sheet to determine if your device has an ePWM module with high-resolution period support.



A. From ePWM Time-base (TB) submodule
B. From ePWM counter-compare (CC) submodule
C. From ePWM Deadband (DB) submodule

Figure 30-108. HRPWM Block Diagram

The ePWM peripheral is used to perform a function mathematically equivalent to a digital-to-analog converter (DAC). As shown in Figure 30-109, the effective resolution for conventionally generated PWM is a function of PWM frequency (or period) and system clock frequency.

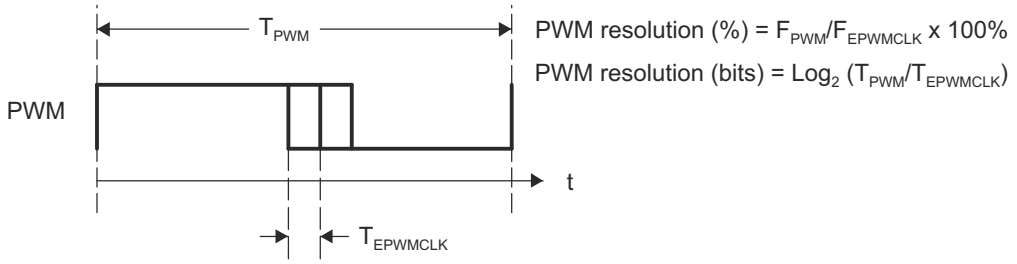


Figure 30-109. Resolution Calculations for Conventionally Generated PWM

If the required PWM operating frequency does not offer sufficient resolution in PWM mode, consider using HRPWM. As an example of improved performance offered by HRPWM, Table 30-16 shows resolution in bits for various PWM frequencies. These values assume a MEP step size of 180ps. See the device data sheet for typical and maximum performance specifications for the MEP.

Table 30-16. Resolution for PWM and HRPWM

PWM Frequency (kHz)	Regular Resolution (PWM) 100MHz EPWMCLK		High Resolution (HRPWM)	
	Bits	%	Bits	%
20	12.3	0.02	18.1	0.000
50	11	0.05	16.8	0.001
100	10	0.1	15.8	0.002
150	9.4	0.15	15.2	0.003
200	9	0.2	14.8	0.004
250	8.6	0.25	14.4	0.005
500	7.6	0.5	13.4	0.009
1000	6.6	1	12.4	0.018
1500	6.1	1.5	11.9	0.027
2000	5.6	2	11.4	0.036

Although each application can differ, typical low-frequency PWM operation (below 250kHz) does not require HRPWM. HRPWM capability is most useful for high-frequency PWM requirements of power conversion topologies such as:

- Single-phase buck, boost, and flyback
- Multiphase buck, boost, and flyback
- Phase-shifted full bridge
- Direct modulation of D-Class power amplifiers

30.18.1 Operational Description of HRPWM

The HRPWM is based on micro-edge positioner (MEP) technology. MEP logic is capable of positioning an edge very finely by sub-dividing one coarse system clock of a conventional PWM generator. The time step accuracy is on the order of 150ps. See the device data sheet for the typical MEP step size on a particular device. The HRPWM also has a self-check software diagnostics mode to check if the MEP logic is running as designed under all operating conditions. Details on software diagnostics and functions are in Section 30.18.1.7.

Figure 30-110 shows the relationship between one coarse system clock and edge position in terms of MEP steps, which are controlled using an 8-bit field in the Compare A extension register (CMPAHR). The same operating logic applies to CMPBHR as well.

To generate an HRPWM waveform, configure the ePWM registers to generate a conventional PWM of a given frequency and polarity. The HRPWM works together with the ePWM registers to extend edge resolution. Although many programming combinations are possible, only a few are needed and practical. These methods are described in Section 30.18.1.8.

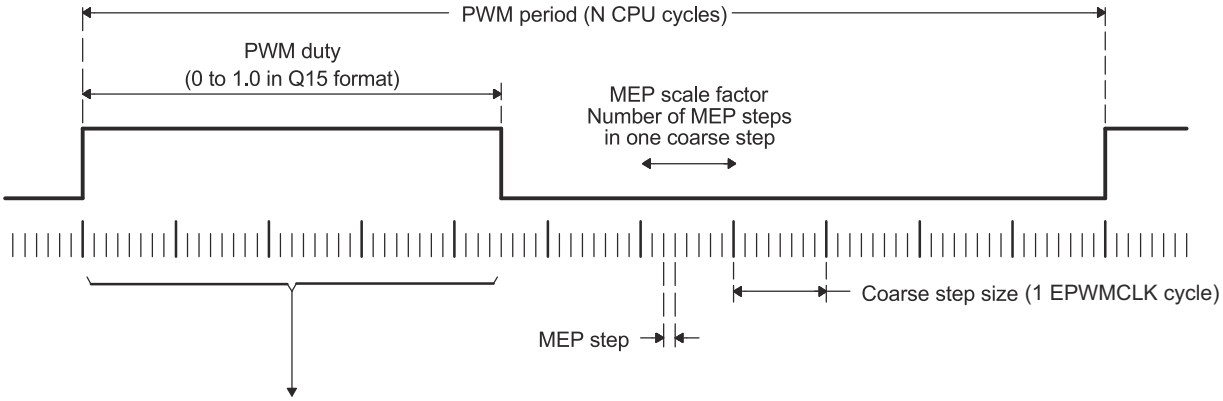
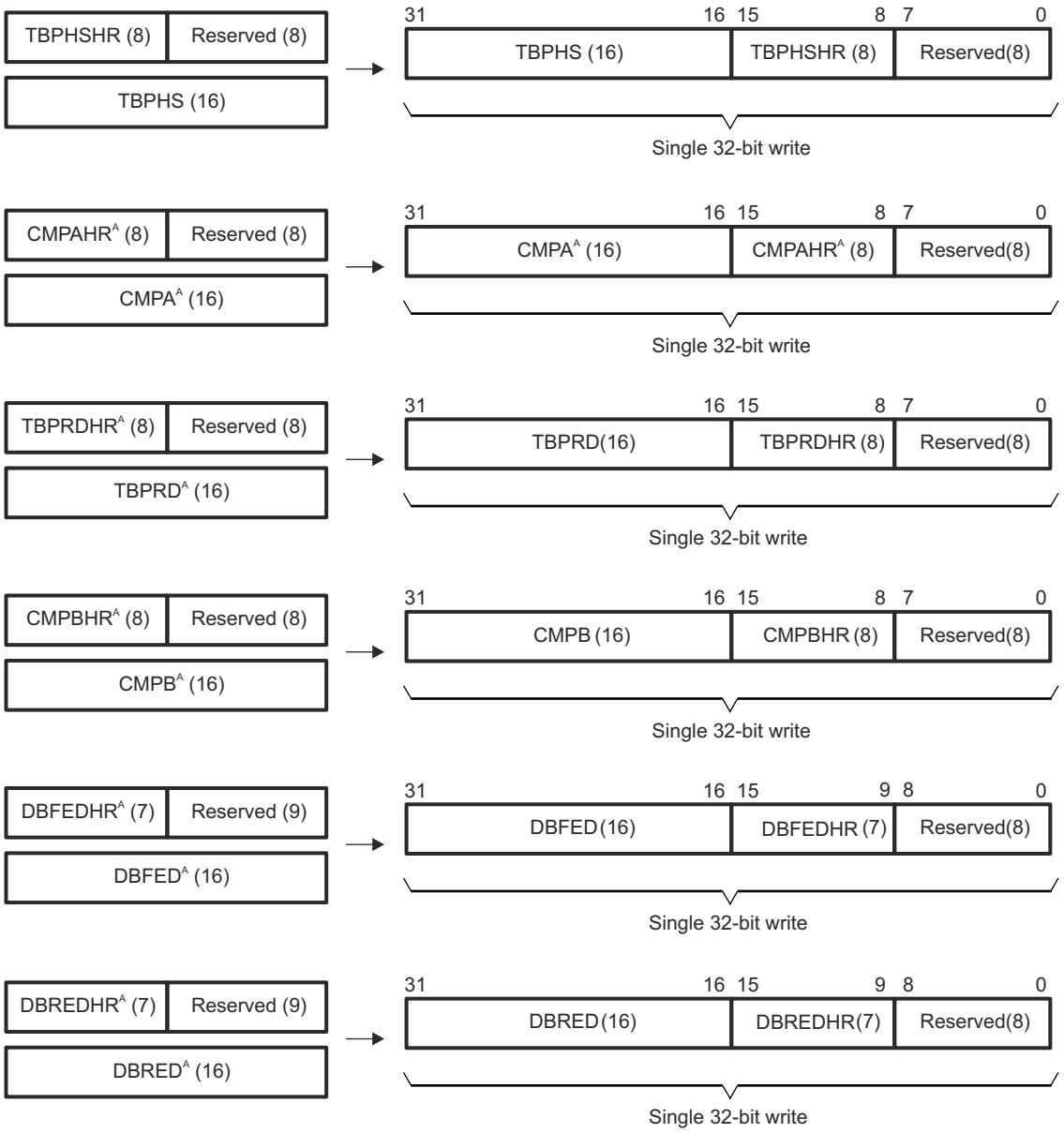


Figure 30-110. Operating Logic Using MEP

30.18.1.1 Controlling the HRPWM Capabilities

The MEP of the HRPWM is controlled by six extension registers. These HRPWM registers are concatenated with the 16-bit TBPHS, TBPRD, CMPA, CMPBM, DBREDM, and DBFEDM registers used to control PWM operation.

- TBPHSHR - Time Base Phase High Resolution Register
- CMPAHR - Counter Compare A High Resolution Register; CMPAHR is for use with the AQ output of Channel A, and is not related to CMPA
- TBPRDHR - Time Base Period High Resolution Register. (available on some devices)
- CMPBHR - Counter Compare B High Resolution Register; CMPBHR is for use with the AQ output of Channel B, and is not related to CMPB
- DBREDHR - Dead-band Generator Rising Edge Delay High Resolution Register
- DBFEDHR - Dead-band Generator Falling Edge Delay High Resolution Register



A. Dependent upon your device, these registers can be mirrored and can be written to at two different memory locations.

Figure 30-111. HRPWM Extension Registers and Memory Configuration

Note

HRPWM capabilities on Deadband Rising Edge Delay and Falling Edge Delay is applicable only during dead band half cycle clocking Operation. The number of MEP steps is half in size [bits 15:9] than duty and phase high-resolution registers for the same reason.

HRPWM capabilities are controlled using the Channel A and B PWM signal path. HRPWM support on the Dead band signal path is available by properly configuring the HRCNFG2 register. shows how the HRPWM interfaces with the 8-bit extension registers.

30.18.1.2 HRPWM Source Clock

Each HRPWM module is clocked from the respective EPWMxCLK. HRCAL has a separate clock. For example, HRPWM1 is sourced from EPWM1CLK while HRPWM2 is clocked from the EPWM2CLK. Figure 30-112 shows the HRCAL and HRPWM modules are sourced from the respective ePWM clock source.

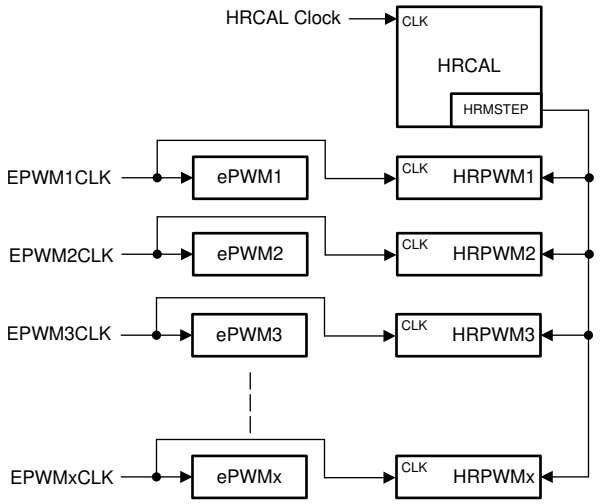


Figure 30-112. HRPWM and HRCAL Source Clock

30.18.1.3 Configuring the HRPWM

Once the ePWM has been configured to provide conventional PWM of a given frequency and polarity, the HRPWM is configured by programming the HRCNFG register in that particular ePWM module's register space. This register provides the following configuration options:

Edge Mode	The MEP can be programmed to provide precise position control on the rising edge (RE), falling edge (FE) or both edges (BE) at the same time. FE and RE are used for power topologies requiring duty cycle control (CMPA or CMPB high-resolution control), while BE is used for topologies requiring phase shifting, for example, phase shifted full bridge (TBPHS or TBPRD high-resolution control).
Control Mode	The MEP is programmed to be controlled either from the CMPAHR/CMPBHR register in case of duty cycle control or the TBPHSHR register (phase control). RE or FE control mode can be used with the CMPAHR or CMPBHR register. BE control mode can be used with the TBPHSHR register. When the MEP is controlled from the TBPRDHR register (period control), the duty cycle and phase can also be controlled using the respective high-resolution registers.
Shadow Mode	This mode provides the same shadowing (double buffering) option as in regular PWM mode. This option is valid only when operating from the CMPAHR, CMPBHR, and TBPRDHR registers and can be chosen to be the same as the regular load option for the CMPA/CMPB register. If TBPHSHR is used, then this option has no effect.
High-Resolution B Signal Control	The B signal path of an ePWM channel can generate a high-resolution output by outputting an inverted version of the high-resolution ePWMxA signal on the ePWMxB pin. A Type 2 or Type 4 HRPWM module can also enable high-resolution features on the B signal path independently of the A signal path as well.
Swap ePWMxA and ePWMxB Outputs	This mode enables the swapping of the high-resolution A and B outputs. The mode selection allows either A and B Outputs Unchanged or A Output Comes Out On B and B Output Comes Out On A.

Auto-conversion Mode

This mode is used in conjunction with the scale factor optimization (SFO) software only. For a type 4 HRPWM module, below is a description of the Auto-conversion Mode taking CMPAHR as an example. If auto-conversion is enabled, CMPAHR = fraction(PWMduty * PWMperiod)<<8. The scale factor optimization software calculates the MEP scale factor in the background code and automatically updates the HRMSTEP register with the calculated number of MEP steps per coarse step. The MEP Calibration Module then uses the values in the HRMSTEP and CMPAHR registers to automatically calculate the appropriate number of MEP steps represented by the fractional duty cycle and moves the high-resolution ePWM signal edge accordingly. If auto-conversion is disabled, the CMPAHR register behaves like a type 0 HRPWM module and CMPAHR = (fraction(PWMduty * PWMperiod) * MEP Scale Factor + 0.5)<<8). All calculations need to be performed by your code in this mode, and the HRMSTEP register is ignored. Auto-conversion for high-resolution period has the same behavior as auto-conversion for high-resolution duty cycle. Auto-conversion must always be enabled for high-resolution period mode.

Note

If the HRPWM module is configured in UP-DOWN counter mode, the shadow mode for the HRPWM registers must be set to load on both ZERO AND PERIOD. New values from the user are loaded to the shadow registers only at CTR = ZERO, but the shadow mode of for the registers must be set to both ZERO AND PERIOD. The CTR = PRD event is used for specific internal logic inside the HRPWM module.

Auto-conversion Mode performs the calculation for CMPBHR , DBREDHR, and DBFEDHR. The scale factor optimization software calculates the MEP scale factor in the background code and automatically updates the HRMSTEP register with the calculated number of MEP steps per coarse step. The MEP Calibration Module then uses the values in the HRMSTEP and CMPBHR or DBREDHR/DBFEDHR register to automatically calculate the appropriate number of MEP steps represented by the fractional components and moves the high-resolution ePWM signal edge accordingly. If auto-conversion is disabled, CMPBHR behaves the same as CMPAHR. CMPBHR = (fraction(PWMduty * PWMperiod) * MEP Scale Factor + 0.5)<<8).

Linking CMPBHR to CMPAHR

Starting with EPWM Type 5, the user has the option to link the CMPBHR value to the CMPAHR value. This allows for EPWM channel A and EPWM channel B outputs to both be controlled by CMPAHR. This feature is enabled through CMPCTL.LINKDUTYHR register. This feature is commonly used when the HRPWM is configured for complimentary output mode.

30.18.1.4 Configuring High-Resolution in Deadband Rising-Edge and Falling-Edge Delay

Once the ePWM has been configured to provide conventional PWM of a given frequency, polarity, and dead band enabled in half-cycle clocking mode, the high-resolution operation on dead band RED and FED lines are enabled by programming the HRCNFG2 register in that particular ePWM module register space. This register provides the following configuration options:

Edge Mode The MEP can be programmed to provide precise position control on the dead band rising edge (RED), dead band falling edge (FED), or both edges (rising edge of DBRED signal and falling edge of DBFED signal) at the same time.

Control Mode Selects the time event that loads the shadow value in the active register for DBRED and DBFED in high-resolution mode. Select the pulse to match the selection in the ePWM DBCTL[LOADREDMODE] and DBCTL[LOADFEDMODE] bits.

30.18.1.5 Principle of Operation

The MEP logic is capable of placing an edge in one of 255 (8 bits) discrete time steps (see the device data sheet for typical MEP step size). The MEP works with the TBM and CCM registers to be certain that time steps are applied and that edge placement accuracy is maintained over a wide range of PWM frequencies, system clock frequencies, and other operating conditions. Table 30-17 shows the typical range of operating frequencies supported by the HRPWM.

Table 30-17. Relationship Between MEP Steps, PWM Frequency, and Resolution

System (MHz)	MEP Steps Per EPWMCLK (1) (2) (3)	PWM Minimum (Hz) (4)	PWM Maximum (MHz)	Resolution at Maximum (Bits) (5)
60.0	93	916	3.00	10.9
70.0	79	1068	3.50	10.6
80.0	69	1221	4.00	10.4
90.0	62	1373	4.50	10.3
100.0	56	1526	5.00	10.1

(1) TBCLK = EPWMCLK.
(2) Table data based on a MEP time resolution of 180ps (this is an example value. See the device data sheet for MEP limits)
(3) MEP steps applied = $T_{EPWMCLK}$/180ps in this example.
(4) PWM minimum frequency is based on a maximum period value,(TBPRD = 65535). PWM mode is asymmetrical up-count.
(5) Resolution in bits is given for the maximum PWM frequency stated.

30.18.1.5.1 Edge Positioning

Note

The following example is presented using the [CMPA:CMPAHR] register combination. The theory of operation and equations are the same, if intending to use the [CMPBM:CMPBHRM] for duty cycle control.

In a typical power control loop, a digital controller issues a duty command, usually expressed in a per unit or percentage terms. Assume that for a particular operating point, the demanded duty cycle is 0.405 or 40.5% on time and the required converter PWM frequency is 1.25MHz. In conventional PWM generation with a system clock of 100MHz, the duty cycle choices are in the vicinity of 40.5%. As shown in Figure 30-113, a compare value of 32 counts (duty = 40%) is the closest to 40.5% that can be attained. This is equivalent to an edge position of 320ns instead of the desired 324ns. This data is shown in Table 30-18.

By utilizing the MEP, an edge position much closer to the desired point of 324ns can be achieved. Table 30-18 shows that in addition to the CMPA value, 22 steps of the MEP (CMPAHR register) positions the edge at 323.96ns, resulting in almost zero error. In this example, assume that the MEP has a step resolution of 180ps.

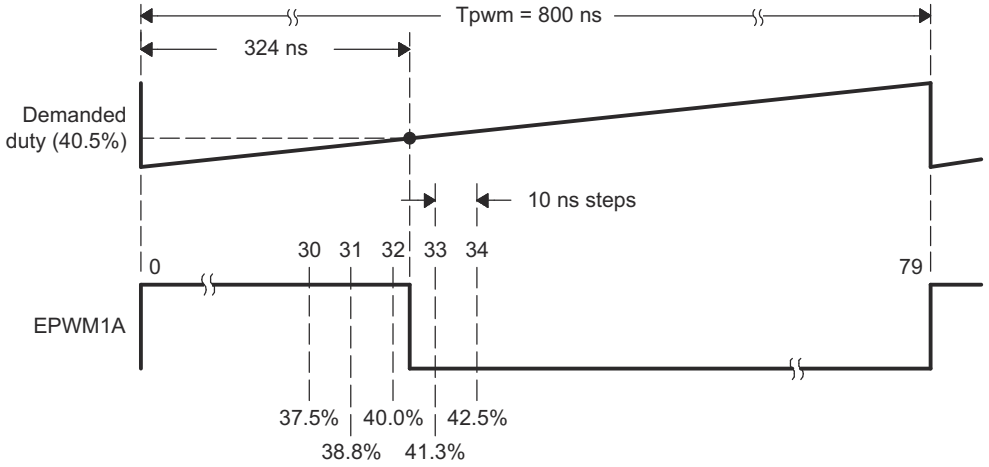


Figure 30-113. Required PWM Waveform for a Requested Duty = 40.5%

Table 30-18. CMPA versus Duty (left), and [CMPA:CMPAHR] versus Duty (right)

CMPA (count)(1) (2) (3)	Duty (%)	High Time (ns)	CMPA (count)	CMPAHR (count)	Duty (%)	High Time (ns)
28	***35.0***	280	32	18	***40.405***	323.24
29	***36.3***	290	32	19	***40.428***	323.42
30	***37.5***	300	32	20	***40.450***	323.60
31	***38.8***	310	32	21	***40.473***	323.78
32	***40.0***	320	32	22	***40.495***	323.96
33	***41.3***	330	32	23	***40.518***	324.14
34	***42.5***	340	32	24	***40.540***	324.32
			32	25	***40.563***	324.50
Required			32	26	***40.585***	324.68
32.40	***40.5***	324	32	27	***40.608***	324.86

(1) Assumed MEP step size for the above example = 180ps. See the device-specific data sheet for typical and maximum MEP values.
(2) TBCLK = 100MHz, 10ns
(3) For a PWM Period register value of 80 counts, PWM Period = 80 × 10ns = 800ns, PWM frequency = 1/800ns = 1.25MHz

30.18.1.5.2 Scaling Considerations

The mechanics of how to position an edge precisely in time has been demonstrated using the resources of the standard CMPA and MEP (CMPAHR) registers. In a practical application, however, it is necessary to seamlessly provide the CPU a mapping function from a per-unit (fractional) duty cycle to a final integer (non-fractional) representation that is written to the [CMPA:CMPAHR] register combination.

To do this, first examine the scaling or mapping steps involved. It is common in control software to express duty cycle in a per-unit or percentage basis. This has the advantage of performing all needed math calculations without concern for the final absolute duty cycle, expressed in clock counts or high time in nanoseconds (ns). Furthermore, it makes the code more transportable across multiple converter types running different PWM frequencies.

To implement the mapping scheme, a two-step scaling procedure is required.

Assumptions for this example:

TBCLK	=	10ns (100MHz)
PWM frequency	=	1.25MHz (1/800ns)
Required PWM duty cycle, **PWMDuty**	=	0.405 (40.5%)
PWM period in terms of coarse steps, **PWMPeriod** (800ns/10ns)	=	80
Number of MEP steps per coarse step at 180ps (10ns/180ps), **MEP_ScaleFactor**	=	55
Value to keep CMPAHR within the range of 1-255 and fractional rounding constant (default value)	=	0.5 (0080h in Q8 format)

Step 1: Percentage Integer Duty value conversion for CMPA register

CMPA register value	=	int(**PWMDuty***PWMPeriod**); int means integer part
	=	int(0.405 * 80)
	=	int(32.4)
CMPA register value	=	32 (20h)

Step 2: Fractional value conversion for CMPAHR register

CMPAHR	=	(frac(**PWMDuty***PWMPeriod**)*MEP_ScaleFactor** + 0.5) <<8); frac means fractional part
	=	(frac(32.4) * 55 + 0.5) <<8; Shifting is to move the value to the high byte of CMPAHR.
	=	(0.4 * 55 + 0.5) <<8
	=	(22 + 0.5) <<8
	=	22.5 * 256; Shifting left by 8 is the same as multiplying by 256.
	=	5760 (1680h)
CMPAHR	=	1680h CMPAHR value = 1600h (lower 8 bits are ignored by hardware).

Note

If the AUTOCONV bit (HRCNFG.6) is set and the MEP_ScaleFactor is in the HRMSTEP register, then CMPAHR / CMPBHR register value = frac (PWMDuty*PWMperiod<<8). The rest of the conversion calculations are performed automatically in hardware, and the correct MEP-scaled signal edge appears on the ePWM channel output. If AUTOCONV is not set, the above calculations must be performed by software.

The MEP scale factor (MEP_ScaleFactor) varies with the system clock and DSP operating conditions. TI provides an MEP scale factor optimizing (SFO) software C function, which uses the built in diagnostics in each HRPWM and returns the best scale factor for a given operating point.

The scale factor varies slowly over a limited range so the optimizing C function can be run very slowly in a background loop.

The CMPA, CMPB, CMPAHR and CMPBHR registers are configured in memory so that the 32-bit data capability of the CPU can write this as a single concatenated value, that is, [CMPA:CMPAHR], [CMPB:CMPBHR], and so on.

The mapping scheme has been implemented in both C and assembly, as shown in Section 30.18.1.8. The actual implementation takes advantage of the 32-bit CPU architecture and is somewhat different from the steps shown in Section 30.18.1.5.2.

For time-critical control loops where every cycle counts, the assembly version is recommended. This is a cycle optimized function (11 EPWMCLK cycles) that takes a Q15 duty value as input and writes a single [CMPA:CMPAHR] value.

30.18.1.5.3 Duty Cycle Range Limitation

In high-resolution mode, the MEP is not active for 100% of the PWM period and becomes operational:

- Three EPWMCLK cycles after the period starts when high-resolution period (TBPRDHR) control is not enabled.
- When high-resolution period (TBPRDHR) control is enabled using the HRPCTL register:
  - In up-count mode: three EPWMCLK cycles after the period starts until three EPWMCLK cycles before the period ends.
  - In up-down count mode: when counting up, three cycles after CTR = 0 until three cycles before CTR = PRD, and when counting down, three cycles after CTR = PRD until three cycles before CTR = 0.
- When using DBREDHR or DBFEDHR, DBRED or DBFED (the register corresponding to the edge with high-resolution displacement) must be greater than or equal to 7.

Duty cycle range limitations are illustrated in Figure 30-114 to Figure 30-117. This limitation imposes a duty cycle limit on the MEP. For example, precision edge control is not available all the way down to 0% duty cycle. When high-resolution period control is disabled, regular PWM duty control is fully operational down to 0% duty cycle despite the unavailability of HRPWM features in the first three cycles. In most applications, this cannot be an issue as the controller regulation point is usually not designed to be close to 0% duty cycle. To better understand the useable duty cycle range, see Table 30-19. When high-resolution period control is enabled (HRPCTL[HRPE] = 1), the duty cycle must not fall within the restricted range; otherwise, there can be undefined behavior on the ePWMxA output.

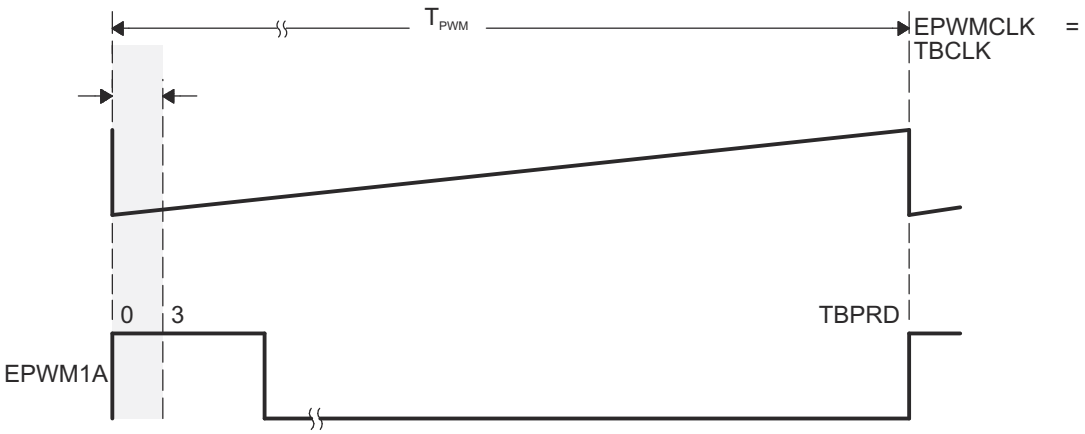


Figure 30-114. Low % Duty Cycle Range Limitation Example (HRPCTL[HRPE] = 0)

Table 30-19. Duty Cycle Range Limitation for Three EPWMCLK/TBCLK Cycles

PWM Frequency(1) (kHz)	3 Cycles Minimum Duty	3 Cycles Maximum Duty(2)
200	0.6%	99.4%
400	1.2%	98.8%
600	1.8%	98.2%
800	2.4%	97.6%
1000	3%	97%
1200	3.6%	96.4%
1400	4.2%	95.8%
1600	4.8%	95.2%
1800	5.4%	94.6%
2000	6%	94%

(1) EPWMCLK = TBCLK = 100MHz
(2) This limitation applies only if high-resolution period (TBPRDHR) control is enabled.

If the application demands HRPWM operation below the minimum duty cycle limitation, then the HRPWM can be configured to operate in count-down mode with the rising edge position (REP) controlled by the MEP when high-resolution period is disabled (HRPCTL[HRPE] = 0). This is illustrated in Figure 30-115. In this configuration, the minimum duty cycle limitation is no longer an issue. However, there is a maximum duty limitation with same percent numbers as given in Table 30-19.

CAUTION

If the application has enabled high-resolution period control (HRPCTL[HRPE] = 1), the duty cycle must not fall within the restricted range; otherwise, there can be undefined behavior on the ePWM output.

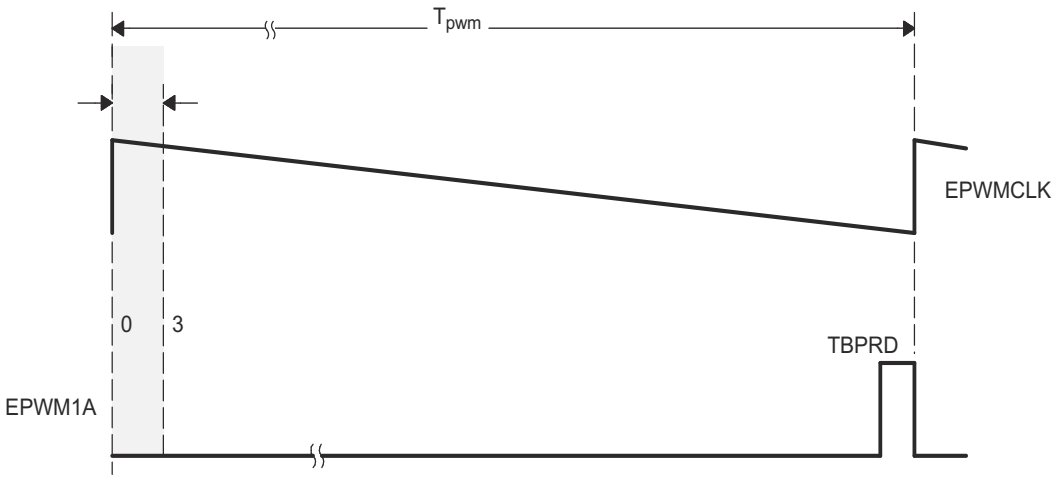


Figure 30-115. High % Duty Cycle Range Limitation Example (HRPCTL[HRPE] = 0)

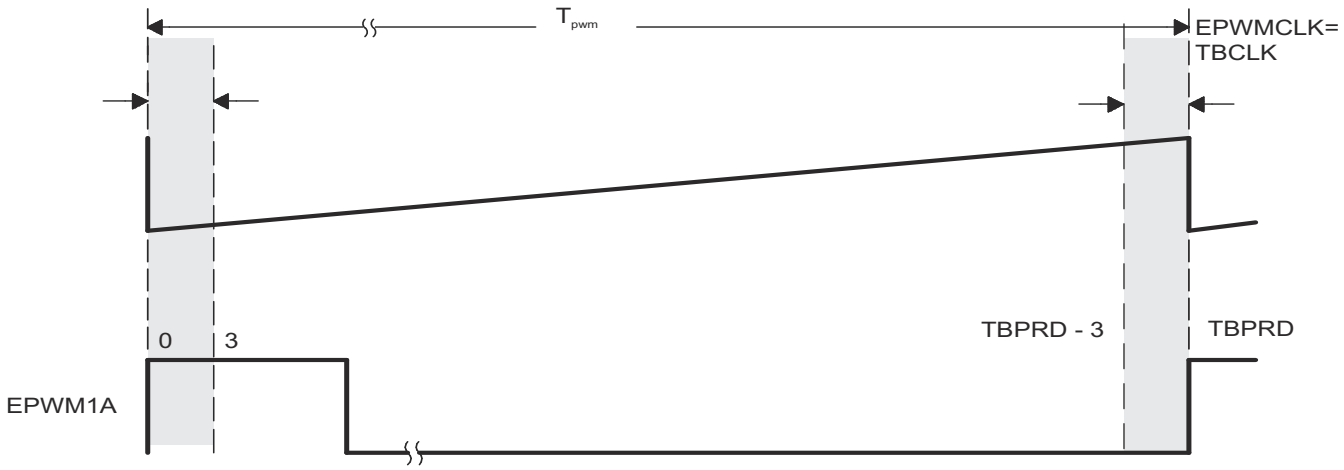


Figure 30-116. Up-Count Duty Cycle Range Limitation Example (HRPCTL[HRPE] = 1)

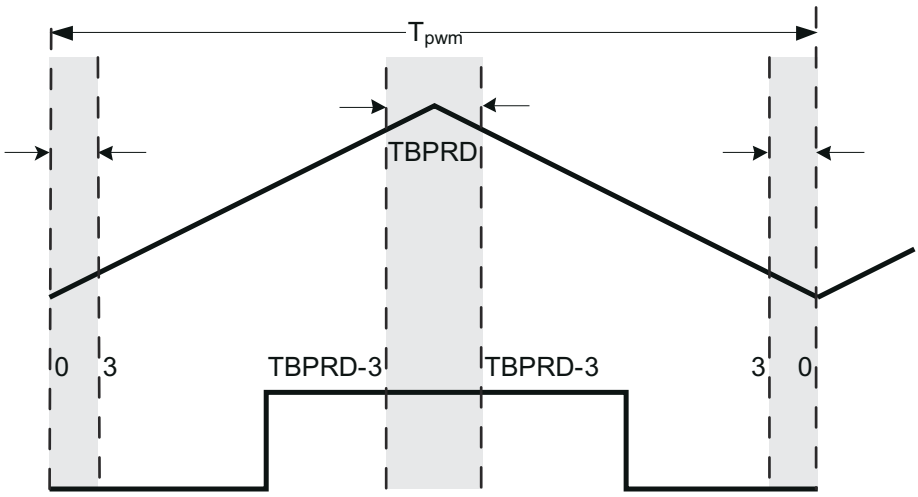


Figure 30-117. Up-Down Count Duty Cycle Range Limitation Example (HRPCTL[HRPE] = 1)

30.18.1.5.4 High-Resolution Period

High-resolution period control using the MEP logic is supported on devices with a Type 1 ePWM module or greater.

Note

When high-resolution period control is enabled, on ePWMxA only, and not ePWMxB output and conversely, the non high-resolution output has ±1 TBCLK cycle jitter in up-count mode and ±2 TBCLK cycle jitter in up-down count mode.

The scaling procedure described for duty cycle in Section 30.18.1.5.2 applies for high-resolution period as well:

Assumptions for this example:

TBCLK	= 10ns (100MHz)
Required PWM frequency	= 175kHz (period of 571.428)
Number of MEP steps per coarse step at 180ps (10ns/180ps), (MEP_ScaleFactor)	= 55
Value to keep TBPRDHR within range of 1-255 and fractional rounding constant (default value)	= 0.5 (0080h in Q8 format)

Problem:

In up-count mode:

- If TBPRD = 571, then PWM frequency = 174.82kHz (period = (571+1)* T_{TBCLK}).
- If TBPRD = 570, then PWM frequency = 175.13kHz (period = (570+1)* T_{TBCLK}).

In up-down count mode:

- If TBPRD = 286, then PWM frequency = 174.82kHz (period = (286*2)* T_{TBCLK}).
- If TBPRD = 285, then PWM frequency = 175.44kHz (period = (285*2)* T_{TBCLK}).

Solution:

With 55 MEP steps per coarse step at 180ps each:

Step 1: Percentage Integer Period value conversion for TBPRD register

Integer period value	= 571 * T_{TBCLK}
	= int (571.428) * T_{TBCLK}
	= int (PWMperiod) * T_{TBCLK}
In up-count mode:	
TBPRD	= 570 (TBPRD = period value - 1)
	= 023Ah
In up-down count mode:	= 285 (TBPRD = period value/2)
TBPRD	= 011Dh

Step 2: Fractional value conversion for TBPRDHR register

In up-count mode:

TBPRDHR register value	= (frac(PWMperiod) * MEP_ScaleFactor + 0.5)
If auto-conversion enabled and HRMSTEP =	
MEP_ScaleFactor value (55):	= frac (PWMperiod) << 8 (Shifting is to move the value to the high byte of TBPRDHR)
TBPRDHR register value	= frac (571.428) << 8
	= 0.428 × 256
	= 6D00h
The auto-conversion then automatically performs the calculation, such that TBPRDHR MEP delay is scaled by hardware to:	= ((TBPRDHR(15:0) >> 8) × HRMSTEP + 80h) << 8
	= (006Dh × 55 + 80h) >> 8
	= (17EBh) >> 8
Period MEP delay	= 0017h MEP Steps

In up-down count mode:

TBPRDHR register value	= (frac(PWMperiod) * MEP_ScaleFactor + 0.5)
If auto-conversion enabled and HRMSTEP =	
MEP_ScaleFactor value (55):	= frac (PWMperiod / 2) << 8 (Shifting is to move the value to the high byte of TBPRDHR)
TBPRDHR register value	= frac (285.714) << 8
	= 0.714 × 256
	= B600h
The auto-conversion then automatically performs the calculation, such that TBPRDHR MEP delay is scaled by hardware to:	= ((TBPRDHR(15:0) >> 8) × HRMSTEP + 80h) << 8
	= (00B6h × 55 + 80h) >> 8
	= (279Ah) >> 8
Period MEP delay	= 0027h MEP Steps

30.18.1.5.4.1 High-Resolution Period Configuration

To use high-resolution period, the ePWMx module must be initialized in the exact order presented.

The following steps use CMPA with shadow registers and the corresponding HRCNFG bits for high-resolution operation on EPWMxA. For high-resolution operation on EPWMxB, make the appropriate substitutions with the B channel fields.

1. Enable ePWMx clock
2. Enable HRPWM clock
3. Disable TBCLKSYNC
4. Configure ePWMx registers - AQ, TBPRD, CC, and so on.
   - ePWMx can only be configured for up-count or up-down count modes. High-resolution period is not compatible with down-count mode.
   - TBPRD and CC registers must be configured for shadow loads.
   - CMPCTL[LOADAMODE]
     - In up-count mode: CMPCTL[LOADAMODE] = 1 (load on CTR = PRD)
     - In up-down count mode: CMPCTL[LOADAMODE] = 2 (load on CTR = 0 or CTR = PRD)
5. Configure the HRCNFG register such that:
   - HRCNFG[HRLOAD] = 2 (load on either CTR = 0 or CTR = PRD)
   - HRCNFG[AUTOCONV] = 1 (Enable auto-conversion)
   - HRCNFG[EDGMODE] = 3 (MEP control on both edges)
6. For TBPHS:TBPHSHR synchronization with high-resolution period, set both HRPCTL[TBPSHRLOADE] = 1 and TBCTL[PHSEN] = 1. In up-down count mode these bits must be set to 1 regardless of the contents of TBPHSHR.
7. Enable high-resolution period control (HRPCTL[HRPE] = 1)
8. Enable TBCLKSYNC
9. TBCTL[SWFSYNC] = 1
10. HRMSTEP must contain an accurate MEP scale factor (# of MEP steps per EPWMCLK coarse step) because auto-conversion is enabled. The MEP scale factor can be acquired using the SFO() function described in Section 30.18.2 .
11. To control high-resolution period, write to the TBPRDHR(M) registers.

Note

When high-resolution period mode is enabled, an EPWMxSYNC pulse introduces ±1-2 cycle jitter to the PWM (±1 cycle in up-count mode and ±2 cycle in up-down count mode). Otherwise, the jitter occurs on every PWM cycle with the synchronization pulse.

When a software synchronization pulse can be issued only once during high-resolution period initialization. If a software sync pulse is applied while the PWM is running, the jitter appears on the PWM output at the time of the sync pulse.

30.18.1.6 Deadband High-Resolution Operation

Note

In up-count mode, the dead-band module is not available when any high-resolution mode is enabled.

Assumptions for this example:

System clock	= 10ns (100MHz)
Deadband enabled in half-cycle mode, TBCLK = EPWMCLK	
Required PWM frequency	1.33MHz (1/750ns)
Required PWM duty cycle	0.5 (50%)
Required Deadband Rising-Edge Delay	5% over duty
Required Deadband Rising-Edge Delay in ns	(0.05 * 375ns) = 18.75ns

Note

Similar to the duty cycle restrictions when using HRPWM, the DBRED and DBFED values must be greater than 3 to use high-resolution deadband.

Deadband delay values as a function of DBFED and DBRED:

When half-cycle clocking is enabled, the formula to calculate the falling-edge delay (FED) and rising-edge delay (RED) becomes:

FED = DBFED * TBCLK / 2

RED = DBRED * TBCLK / 2

DBRED and DBFED calculated values:

Required Deadband Rising-Edge Delay in ns = 18.75ns

DBRED = RED / (TBCLK / 2)

DBRED = 18.75ns/5ns

DBRED Required = 3.75ns

With 55 MEP steps per coarse step at 180ps each:

Step 1: Integer Deadband value conversion for DBREDM register

Integer DBRED value	= int (RED / (TBCLK / 2))
	= int (3.75)
DBRED	= 3

Step 2: Fractional value conversion for Deadband high-resolution register DBREDHR

DBREDHR register value	= (frac(DBRED Required) * MEP_ScaleFactor + 0.5) << 8 (Shifting is to move the value to the high byte of DBREDHR)
	= (frac (3.75) * 55 + 0.5) << 8
	= (0.75 * 55 + 0.5) << 8
	= (41.75) * 256 Shifting left by 8 is the same as multiplying by 256.
DBREDHR value	= 29C0h MEP Steps
	Hardware ignores lower 9 bits in the above calculated DBREDHR value

Note

If the AUTOCONV bit (HRCNFG.6) is set and the MEP_ScaleFactor is in the HRMSTEP register, then DBREDHR:DBRED = frac((required DB value) < <8). The rest of the conversion calculations are performed automatically in hardware, and the correct MEP-scaled signal edge appears on the ePWM channel output. If AUTOCONV is not set, the above calculations must be performed by software.

30.18.1.7 Scale Factor Optimizing Software (SFO)

The micro edge positioner (MEP) logic is capable of placing an edge in one of 255 discrete time steps. As previously mentioned, the size of these steps is on the order of 150ps (see the device data sheet for typical MEP step size on your device). The MEP step size varies based on worst-case process parameters, operating temperature, and voltage. MEP step size increases with decreasing voltage and increasing temperature and decreases with increasing voltage and decreasing temperature. Applications that use the HRPWM feature can use the TI-supplied MEP scale factor optimization (SFO) software function. The SFO function helps to dynamically determine the number of MEP steps per EPWMCLK period while the HRPWM is in operation.

To utilize the MEP capabilities effectively, the correct value for the MEP scaling factor needs to be known by the software. To accomplish this, the HRPWM module has built in self-check and diagnostic capabilities that can be used to determine the optimum MEP scale factor value for any operating condition. TI provides a C-callable library containing one SFO function that utilizes this hardware and determines the optimum MEP scale factor. As such, MEP control and diagnostics registers are reserved for TI use.

A detailed description of the SFO library - SFO_TI_Build_V8.lib software can be found in SFO Library Software - SFO TI_Build_V8.lib.

30.18.1.8 HRPWM Examples Using Optimized Assembly Code

The best way to understand how to use the HRPWM capabilities is through two real examples:

1. Simple buck converter using asymmetrical PWM (count-up) with active high polarity.
2. DAC function using simple R+C reconstruction filter.

The following examples all have initialization and configuration code written in C. To make these easier to understand, the #defines shown below are used.

Example 30-2 assumes MEP step size of 150ps and does not use the SFO library.

Example 30-2. #Defines for HRPWM Header Files

```
// HRPWM (High Resolution PWM) //
================================
// HRCNFG
#define HR_Disable 0x0
#define HR_REP 0x1          // Rising Edge position
#define HR_FEP 0x2          // Falling Edge position
#define HR_BEP 0x3          // Both Edge position #define HR_CMP 0x0 // CMPAHR controlled
#define HR_PHS 0x1          // TBPHSHR controlled #define HR_CTR_ZERO 0x0 // CTR = Zero event
#define HR_CTR_PRD 0x1      // CTR = Period event
#define HR_CTR_ZERO_PRD 0x2 // CTR = ZERO or Period event
#define HR_NORM_B  0x0      // Normal ePWMxB output
#define HR_INVERT_B 0x1     // ePWMxB is inverted ePWMxA output
```

30.18.1.8.1 Implementing a Simple Buck Converter

In this example, the PWM requirements are:

- PWM frequency = 1MHz (that is, TBPRD = 100)
- PWM mode = asymmetrical, up-count
- Resolution = 12.7 bits (with a MEP step size of 150ps)

Figure 30-118 and Figure 30-119 show the required PWM waveform. As explained previously, configuration for the ePWM1 module is almost identical to the normal case except that the appropriate MEP options need to be enabled/selected.

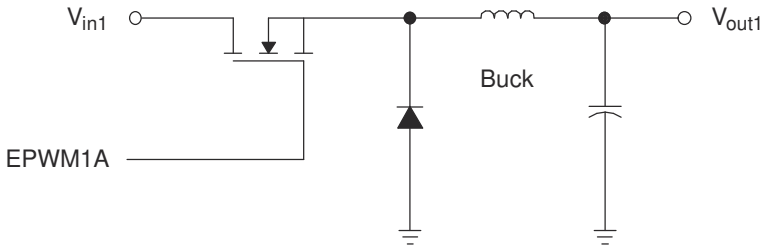


Figure 30-118. Simple Buck Controlled Converter Using a Single PWM

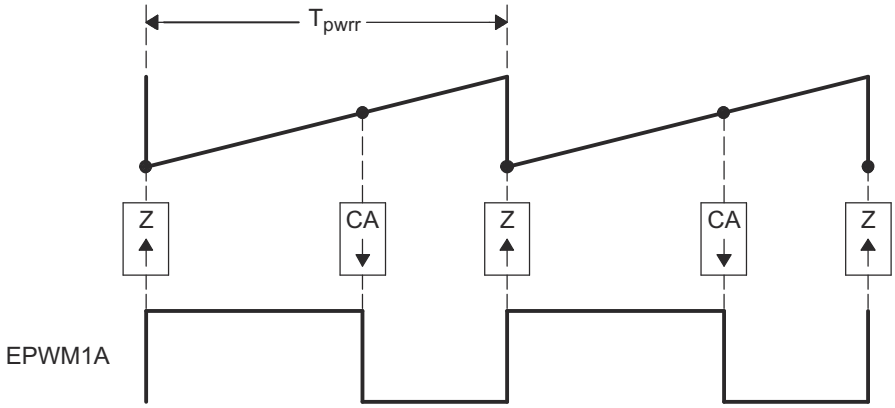


Figure 30-119. PWM Waveform Generated for Simple Buck Controlled Converter

The example code shown consists of two main parts:

- Initialization code (executed once)
- Run time code (typically executed within an ISR)

Example 30-3 shows the Initialization code. The first part is configured for conventional PWM. The second part sets up the HRPWM resources.

This example assumes MEP step size of 150ps and does not use the SFO library.

Example 30-4 shows an assembly example of run-time code for the HRPWM buck converter.

Example 30-3. HRPWM Buck Converter Initialization Code

```
void HrBuckDrvCnf(void)
 {
// Config for conventional PWM first
EPwm1Regs.TBCTL.bit.PRDLD = TB_IMMEDIATE;         // set Immediate load
EPwm1Regs.TBPRD = 100;                            // Period set for 1000kHz PWM
hrbuck_period = 200;                              // Used for Q15 to Q0 scaling
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;           // EPWM1 is the Sync Source

EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
// Note: ChB is initialized here only for comparison purposes, it is not required

EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;     // optional
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;       // optional
EPwm1Regs.AQCTLA.bit.ZRO = AQ_SET;
EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_SET;                // optional
EPwm1Regs.AQCTLB.bit.CBU = AQ_CLEAR;              // optional
// Now configure the HRPWM resources
EALLOW;                                           // Note these registers are protected
                                                  // and act only on ChA
EPwm1Regs.HRCNFG.all = 0x0;                       // clear all bits first
EPwm1Regs.HRCNFG.bit.EDGMODE = HR_FEP;            // Control Falling Edge Position
EPwm1Regs.HRCNFG.bit.CTLMODE = HR_CMP;            // CMPAHR controls the MEP
EPwm1Regs.HRCNFG.bit.HRLOAD = HR_CTR_ZERO;        // Shadow load on CTR=Zero
EDIS;
MEP_ScaleFactor = 66*256;                         // Start with typical Scale Factor
                                                  // value for 100MHz
                                                  // Note: Use SFO functions to update
                                                     MEP_ScaleFactor dynamically
}
```

Example 30-4. HRPWM Buck Converter Run-Time Code

```
EPWM1_BASE .set 0x6800
CMPAHR1 .set EPWM1_BASE+0x8
;=============================================
HRBUCK_DRV; (can execute within an ISR or loop)
;=============================================
     MOVW DP, #_HRBUCK_In
     MOVL XAR2,@_HRBUCK_In       ; Pointer to Input Q15 Duty (XAR2)
     MOVL XAR3,#CMPAHR1          ; Pointer to HRPWM CMPA reg (XAR3)

; Output for EPWM1A (HRPWM)
     MOV T,*XAR2 ; T <= Duty
     MPYU ACC,T,@_hrbuck_period  ; Q15 to Q0 scaling based on Period
     MOV T,@_MEP_ScaleFactor     ; MEP scale factor (from optimizer s/w)
     MPYU P,T,@AL                ; P <= T * AL, Optimizer scaling
     MOVH @AL,P                  ; AL <= P, move result back to ACC
     ADD ACC, #0x080             ; MEP range and rounding adjustment
     MOVL *XAR3,ACC              ; CMPA:CMPAHR(31:8) <= ACC

; Output for EPWM1B (Regular Res) Optional - for comparison purpose only
     MOV *+XAR3[2],AH            ; Store ACCH to regular CMPB
```

30.18.1.8.2 Implementing a DAC Function Using an R+C Reconstruction Filter

In this example, the PWM requirements are:

- PWM frequency = 400kHz (that is, TBPRD = 250)
- PWM mode = Asymmetrical, Up-count
- Resolution = 14 bits (MEP step size = 150ps)

Figure 30-120 and Figure 30-121 show the DAC function and the required PWM waveform. As explained previously, configuration for the ePWM1 module is almost identical to the normal case except that the appropriate MEP options need to be enabled/selected.

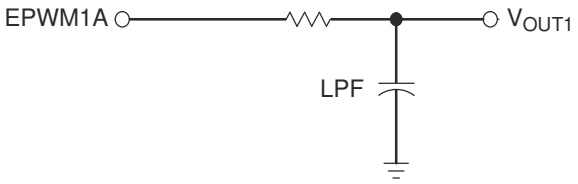


Figure 30-120. Simple Reconstruction Filter for a PWM-based DAC

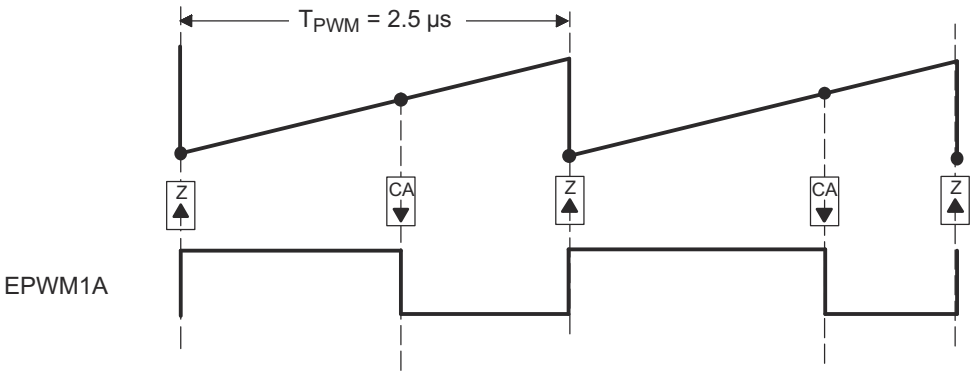


Figure 30-121. PWM Waveform Generated for the PWM DAC Function

The example code shown consists of two main parts:

- Initialization code (executed once)
- Run time code (typically executed within an ISR)

This example assumes a typical MEP_SP and does not use the SFO library.

Example 30-5 shows the Initialization code. The first part is configured for conventional PWM. The second part sets up the HRPWM resources.

Example 30-6 shows an assembly example of run-time code that can execute in a high-speed ISR loop.

Example 30-5. PWM DAC Function Initialization Code

```
void HrPwmDacDrvCnf(void)
{
// Config for conventional PWM first
EPwm1Regs.TBCTL.bit.PRDLD = TB_IMMEDIATE;      // Set Immediate load
EPwm1Regs.TBPRD = 250;                         // Period set for 400kHz PWM
hrDAC_period = 250;                            // Used for Q15 to Q0 scaling
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;        // EPWM1 is the Sync Source

EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
// Note: ChB is initialized here only for comparison purposes, it is not required

EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;  // optional
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;    // optional

EPwm1Regs.AQCTLA.bit.ZRO = AQ_SET;
EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_SET;             // optional
EPwm1Regs.AQCTLB.bit.CBU = AQ_CLEAR;           // optional
// Now configure the HRPWM resources
EALLOW;                                        // Note these registers are protected
                                               // and act only on ChA.
EPwm1Regs.HRCNFG.all = 0x0;                    // Clear all bits first
EPwm1Regs.HRCNFG.bit.EDGMODE = HR_FEP;         // Control falling edge position
EPwm1Regs.HRCNFG.bit.CTLMODE = HR_CMP;         // CMPAHR controls the MEP.
EPwm1Regs.HRCNFG.bit.HRLOAD = HR_CTR_ZERO;     // Shadow load on CTR=Zero.
EDIS;
MEP_ScaleFactor = 66*256;                      // Start with typical Scale Factor
                                               // value for 100MHz.
                                               // Use SFO functions to update MEP_ScaleFactor
                                               // dynamically.
}
```

Example 30-6. PWM DAC Function Run-Time Code

```
EPWM1_BASE .set 0x6800
CMPAHR1 .set EPWM1_BASE+0x8
;================================================
HRPWM_DAC_DRV; (can execute within an ISR or loop)
;================================================
      MOVW DP, #_HRDAC_In
      MOVL XAR2,@_HRDAC_In             ; Pointer to input Q15 duty (XAR2)
      MOVL XAR3,#CMPAHR1               ; Pointer to HRPWM CMPA reg (XAR3)

; Output for EPWM1A (HRPWM
      MOV T,*XAR2                      ; T <= duty
      MPY ACC,T,@_hrDAC_period         ; Q15 to Q0 scaling based on period
      ADD ACC,@_HrDAC_period<<15       ; Offset for bipolar operation
      MOV T,@_MEP_ScaleFactor          ; MEP scale factor (from optimizer s/w)
      MPYU P,T,@AL                     ; P <= T * AL, optimizer scaling
      MOVH @AL,P                       ; AL <= P, move result back to ACC
      ADD ACC, #0x080                  ; MEP range and rounding adjustment
      MOVL *XAR3,ACC                   ; CMPA:CMPAHR(31:8) <= ACC

; Output for EPWM1B (Regular Res) Optional - for comparison purpose only
      MOV *+XAR3[2],AH                 ; Store ACCH to regular CMPB
```

30.18.2 SFO Library Software - SFO_TI_Build_V8.lib

Table 30-20 lists several features of the SFO_TI_Build_V8.lib library.

Table 30-20. SFO Library Features

	SFO_TI_Build_V8.lib	Unit
Completion-checking?	Yes	Function return value
Typical cycles required for SFO() to update MEP_ScaleFactor if called repetitively without interrupts	130,000	EPWMCLK cycles

30.18.2.1 Scale Factor Optimizer Function - int SFO()

This routine drives the micro-edge positioner (MEP) calibration module to run SFO diagnostics and determine the appropriate MEP scale factor (number of MEP steps per coarse EPWMCLK step) for a device at any given time.

If EPWMCLK = TBCLK = 100MHz and assuming the MEP step size is 150ps, the typical scale factor value at 100MHz = 66 MEP steps per TBCLK unit (10ns)

The function returns a MEP scale factor value:

MEP_ScaleFactor = Number of MEP steps per EPWMCLK

Constraints when using this function:

- SFO() can be used with a minimum EPWMCLK = TBCLK = 50MHz. MEP diagnostics logic uses EPWMCLK and not TBCLK, so the EPWMCLK restriction is an important constraint. Below 50MHz with device process variation, the MEP step size can decrease under cold temperature and high core voltage conditions to such a point that 255 MEP steps do not span an entire EPWMCLK cycle.
- At any time, SFO() can be called to run SFO diagnostics on the MEP calibration module.

Usage:

- SFO() can be called at any time in the background while the ePWM channels are running in HRPWM mode. The scale factor result obtained can be applied to all ePWM channels running in HRPWM mode because the function makes use of the diagnostics logic in the MEP calibration module (which runs independently of ePWM channels).
- This routine returns a 1 when calibration is finished and a new scale factor has been calculated or returns a 0 if calibration is still running. The routine returns a 2 if there is an error, and the MEP_ScaleFactor is greater than the maximum 255 fine steps per coarse EPWMCLK cycle. In this case, the HRMSTEP register maintains the last MEP scale factor value less than 256 for auto conversion.
- All ePWM modules operating in HRPWM incur only a 3 EPWMCLK cycle minimum duty cycle limitation when high-resolution period control is not used. If high-resolution period control is enabled, there is an additional duty cycle limitation 3-EPWMCLK cycles before the end of the PWM period (see Section 30.18.1.5.3).
- The SFO() function also updates the HRMSTEP register with the scale factor result. If the HRCNFG[AUTOCONV] bit is set, the application software is responsible only for setting CMPAHR = fraction(PWMduty*PWMperiod) << 8 or CMPBHR = fraction(PWMduty*PWMperiod) << 8 or TBPRDHR = fraction (PWMperiod) while running SFO() in the background. The MEP Calibration Module then uses the values in the HRMSTEP and CMPAHR/CMPBHR/TBPRDHR register to automatically calculate the appropriate number of MEP steps represented by the fractional duty cycle or period and move the high-resolution ePWM signal edge accordingly.
- If the HRCNFG[AUTOCONV] bit is clear, the HRMSTEP register is ignored. The application software needs to perform the necessary calculations manually so that:
  - CMPAHR = (fraction(PWMduty * PWMperiod) * MEP Scale Factor) << 8 + 0x080.
  - Similar behavior applies for TBPHSHR, CMPBHR, DBREDHR, and DBFEDHR. Auto-conversion must be enabled when using TBPRDHR.

The following code snippet shows how to use the HRPWM DUTY using driverlib functions.

```
float32_t dutyFine = 85.62;
float32_t count = (dutyFine * (float32_t)(EPWM_TIMER_TBPRD << 8))/100;
uint32_t compCount = (count);
HRPWM_setCounterCompareValue(EPWM1_BASE, HRPWM_COUNTER_COMPARE_A, compCount);
HRPWM_setCounterCompareValue(EPWM1_BASE, HRPWM_COUNTER_COMPARE_B, compCount);
```

The routine can be run as a background task in a slow loop requiring negligible CPU cycles. The repetition rate at which an SFO function needs to be executed depends on the application's operating environment. As with all digital CMOS devices, temperature and supply voltage variations have an effect on MEP operation. However, in most applications these parameters vary slowly and therefore is often sufficient to execute the SFO function once every 5 to 10 seconds. If more rapid variations are expected, then execution can be performed more frequently to match the application. Note there is no high limit restriction on the SFO function repetition rate; hence, the SFO function can execute as quickly as the background loop is capable.

While using the HRPWM feature, HRPWM logic is not active for the first 3 EPWMCLK cycles of the PWM period (and the last 3 EPWMCLK cycles of the PWM period if TBPRDHR is used). While running the application in this configuration, if high-resolution period control is disabled (HRPCTL[HRPE = 0]) and the CMPA/CMPB register value is less than 3 cycles, then the CMPAHR/CMPBHR register must be cleared to zero. If high-resolution period control is enabled (HRPCTL[HRPE = 1]), the CMPA register value must not fall below 3 or above TBPRD-3.This can avoid any unexpected transitions on the PWM signal.

30.18.2.2 Software Usage

The software library function SFO(), calculates the MEP scale factor for the HRPWM-supported ePWM modules. The scale factor is an integer value in the range 1-255, and represents the number of micro step edge positions available for a system clock period. The scale factor value is returned in an integer variable called MEP_ScaleFactor. For example, see Table 30-21.

Table 30-21. Factor Values

Software Function call	Functional Description	Updated Variables
SFO()	Returns MEP scale factor in the HRMSTEP register	MEP_ScaleFactor and HRMSTEP register.

To use the HRPWM feature of the ePWMs, it is recommended that the SFO function be used as described here.

Step 1. Add "Include" Files

The SFO_V8.h file needs to be included as follows. This include file is mandatory while using the SFO library function. For the SFO() to operate, the appropriate (Device)_Device.h and (Device)_Epwm_defines.h must be included in the project. These include files are optional if customized header files are used in the end applications.

Example 30-7. A Sample of How to Add "Include" Files

```
#include "F28x7x_Device.h"         // F28x7x Headerfile
#include "F28x7x_EPwm_defines.h"   // init defines
#include "SFO_V8.h"                // SFO lib functions (needed for HRPWM)
```

Step 2. Element Declaration

Declare an integer variable for the scale factor value as shown below.

Example 30-8. Declaring an Element

```
int MEP_ScaleFactor = 0;   //scale factor value
volatile struct EPWM_REGS *ePWM[] = {0, &EPwm1Regs, &EPwm2Regs, &EPwm3Regs,
&EPwm4Regs};
```

Step 3. MEP_ScaleFactor Initialization

The SFO() function does not require a starting scale factor value in MEP_ScaleFactor. Prior to using the MEP_ScaleFactor variable in application code, SFO() can be called to drive the MEP calibration module to calculate an MEP_ScaleFactor value.

As part of the one-time initialization code prior to using MEP_ScaleFactor, include the following:

Example 30-9. Initializing With a Scale Factor Value

```
MEP_ScaleFactor initialized using function SFO ()
while (SFO() == 0) {} // MEP_ScaleFactor calculated by MEP Cal Module
```

Step 4. Application Code

While the application is running, fluctuations in both device temperature and supply voltage can be expected. To be sure that good Scale Factors are used for each ePWM module, the SFO function can be re-run periodically as part of a slower back-ground loop. Some examples of this are shown here.

Note

See the HRPWM_SFO example in the device-specific C/C++ header files and peripheral examples available from the TI website.

Example 30-10. SFO Function Calls

```
main ()
 {
 int status;
    // User code
    // ePWM1, 2, 3, 4 are running in HRPWM mode
    // The status variable returns 1 once a new MEP_ScaleFactor has been
    // calculated by the MEP Calibration Module running SFO
    // diagnostics.
status = SFO();
if(status==2) {ESTOP0;}   // The function returns a 2 if MEP_ScaleFactor is greater
                          // than the maximum 255 allowed (error condition)
 }
```

30.19 Software

30.19.1 EPWM Registers to Driverlib Functions

Table 30-22. EPWM Registers to Driverlib Functions

File	Driverlib Function
XCMPCTL1	
epwm.h	EPWM_enableXCMPMode
epwm.h	EPWM_disableXCMPMode
epwm.h	EPWM_enableSplitXCMP
epwm.h	EPWM_disableSplitXCMP
epwm.h	EPWM_allocAXCMP
epwm.h	EPWM_allocBXCMP
XLOADCTL	
epwm.h	EPWM_setXCMPLoadMode
epwm.h	EPWM_setXCMPShadowLevel
epwm.h	EPWM_setXCMPShadowBufPtrLoadOnce
epwm.h	EPWM_setXCMPShadowRepeatBufxCount
XLOAD	
epwm.h	EPWM_enableXLoad
epwm.h	EPWM_forceXLoad
epwm.h	EPWM_setXCMPLoadMode
epwm.h	EPWM_setXCMPShadowLevel
epwm.h	EPWM_setXCMPShadowBufPtrLoadOnce
epwm.h	EPWM_setXCMPShadowRepeatBufxCount
XLINKXLOAD	
-	
XREGSHDW1STS	
-	
XREGSHDW2STS	
-	
XREGSHDW3STS	
-	
XCMP1_ACTIVE	
epwm.h	EPWM_setXCMPRegValue
hrpwm.h	HRPWM_setXCMPRegValue
hrpwm.h	HRPWM_setHiResXCMPRegValueOnly
XCMP2_ACTIVE	
-	
XCMP3_ACTIVE	
-	
XCMP4_ACTIVE	
-	
XCMP5_ACTIVE	
-	
XCMP6_ACTIVE	
-	
XCMP7_ACTIVE	

Table 30-22. EPWM Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
XCMP8_ACTIVE	
-	
XTBPRD_ACTIVE	
-	
XAQCTLA_ACTIVE	
epwm.h	EPWM_setXCMPActionQualifierAction
XAQCTLB_ACTIVE	
-	
XMINMAX_ACTIVE	
epwm.h	EPWM_setXMINMAXRegValue
XCMP1_SHDW1	
-	
XCMP2_SHDW1	
-	
XCMP3_SHDW1	
-	
XCMP4_SHDW1	
-	
XCMP5_SHDW1	
-	
XCMP6_SHDW1	
-	
XCMP7_SHDW1	
-	
XCMP8_SHDW1	
-	
XTBPRD_SHDW1	
-	
XAQCTLA_SHDW1	
epwm.h	EPWM_setXCMPActionQualifierAction
XAQCTLB_SHDW1	
-	
CMPC_SHDW1	
epwm.h	EPWM_setCMPShadowRegValue
CMPD_SHDW1	
-	
XMINMAX_SHDW1	
-	
XCMP1_SHDW2	
-	
XCMP2_SHDW2	
-	
XCMP3_SHDW2	
-	

Table 30-22. EPWM Registers to Driverlib Functions (continued)

File	Driverlib Function
XCMP4_SHDW2	
-	
XCMP5_SHDW2	
-	
XCMP6_SHDW2	
-	
XCMP7_SHDW2	
-	
XCMP8_SHDW2	
-	
XTBPRD_SHDW2	
-	
XAQCTLA_SHDW2	
epwm.h	EPWM_setXCMPActionQualifierAction
XAQCTLB_SHDW2	
-	
CMPC_SHDW2	
-	
CMPD_SHDW2	
-	
XMINMAX_SHDW2	
-	
XCMP1_SHDW3	
-	
XCMP2_SHDW3	
-	
XCMP3_SHDW3	
-	
XCMP4_SHDW3	
-	
XCMP5_SHDW3	
-	
XCMP6_SHDW3	
-	
XCMP7_SHDW3	
-	
XCMP8_SHDW3	
-	
XTBPRD_SHDW3	
-	
XAQCTLA_SHDW3	
epwm.h	EPWM_setXCMPActionQualifierAction
XAQCTLB_SHDW3	
-	
CMPC_SHDW3	

Table 30-22. EPWM Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
CMPD_SHDW3	
-	
XMINMAX_SHDW3	
-	
TBCTL	
epwm.c	EPWM_setEmulationMode
epwm.h	EPWM_setCountModeAfterSync
epwm.h	EPWM_setClockPrescaler
epwm.h	EPWM_forceSyncPulse
epwm.h	EPWM_setOneShotSyncOutTrigger
epwm.h	EPWM_setPeriodLoadMode
epwm.h	EPWM_enablePhaseShiftLoad
epwm.h	EPWM_disablePhaseShiftLoad
epwm.h	EPWM_setTimeBaseCounterMode
epwm.h	EPWM_selectPeriodLoadEvent
epwm.h	EPWM_enableOneShotSync
epwm.h	EPWM_disableOneShotSync
epwm.h	EPWM_startOneShotSync
TBCTL2	
epwm.h	EPWM_selectPeriodLoadEvent
epwm.h	EPWM_enableOneShotSync
epwm.h	EPWM_disableOneShotSync
epwm.h	EPWM_startOneShotSync
SYNCINSEL	
epwm.h	EPWM_setSyncInPulseSource
TBCTR	
epwm.h	EPWM_setTimeBaseCounter
epwm.h	EPWM_getTimeBaseCounterValue
TBSTS	
epwm.h	EPWM_getTimeBaseCounterOverflowStatus
epwm.h	EPWM_clearTimeBaseCounterOverflowEvent
epwm.h	EPWM_getSyncStatus
epwm.h	EPWM_clearSyncEvent
epwm.h	EPWM_getTimeBaseCounterDirection
SYNCOUTEN	
epwm.h	EPWM_enableSyncOutPulseSource
epwm.h	EPWM_disableSyncOutPulseSource
TBCTL3	
epwm.h	EPWM_setOneShotSyncOutTrigger
CMPCTL	
epwm.h	EPWM_setCounterCompareShadowLoadMode
epwm.h	EPWM_disableCounterCompareShadowLoadMode
epwm.h	EPWM_getCounterCompareShadowStatus
epwm.h	EPWM_enableLinkDutyHR

Table 30-22. EPWM Registers to Driverlib Functions (continued)

File	Driverlib Function
epwm.h	EPWM_disableLinkDutyHR
CMPCTL2	
epwm.h	EPWM_setCounterCompareShadowLoadMode
epwm.h	EPWM_disableCounterCompareShadowLoadMode
DBCTL	
epwm.h	EPWM_setDeadBandOutputSwapMode
epwm.h	EPWM_setDeadBandDelayMode
epwm.h	EPWM_setDeadBandDelayPolarity
epwm.h	EPWM_setRisingEdgeDeadBandDelayInput
epwm.h	EPWM_setFallingEdgeDeadBandDelayInput
epwm.h	EPWM_setDeadBandControlShadowLoadMode
epwm.h	EPWM_disableDeadBandControlShadowLoadMode
epwm.h	EPWM_setRisingEdgeDelayCountShadowLoadMode
epwm.h	EPWM_disableRisingEdgeDelayCountShadowLoadMode
epwm.h	EPWM_setFallingEdgeDelayCountShadowLoadMode
epwm.h	EPWM_disableFallingEdgeDelayCountShadowLoadMode
epwm.h	EPWM_setDeadBandCounterClock
DBCTL2	
epwm.h	EPWM_setDeadBandControlShadowLoadMode
epwm.h	EPWM_disableDeadBandControlShadowLoadMode
AQCTL	
epwm.h	EPWM_setActionQualifierShadowLoadMode
epwm.h	EPWM_disableActionQualifierShadowLoadMode
epwm.h	EPWM_setActionQualifierAction
epwm.h	EPWM_setActionQualifierActionComplete
epwm.h	EPWM_setAdditionalActionQualifierActionComplete
AQTSRCSEL	
epwm.h	EPWM_setActionQualifierT1TriggerSource
epwm.h	EPWM_setActionQualifierT2TriggerSource
PCCTL	
epwm.h	EPWM_enableChopper
epwm.h	EPWM_disableChopper
epwm.h	EPWM_setChopperDutyCycle
epwm.h	EPWM_setChopperFreq
epwm.h	EPWM_setChopperFirstPulseWidth
VCAPCTL	
epwm.h	EPWM_enableValleyCapture
epwm.h	EPWM_disableValleyCapture
epwm.h	EPWM_startValleyCapture
epwm.h	EPWM_setValleyTriggerSource
epwm.h	EPWM_enableValleyHWDelay
epwm.h	EPWM_disableValleyHWDelay
epwm.h	EPWM_setValleyDelayDivider
VCNTCFG	
epwm.h	EPWM_setValleyTriggerEdgeCounts

Table 30-22. EPWM Registers to Driverlib Functions (continued)

File	Driverlib Function
epwm.h	EPWM_getValleyEdgeStatus
HRCNFG	
hrpwm.h	HRPWM_setMEPEdgeSelect
hrpwm.h	HRPWM_setMEPControlMode
hrpwm.h	HRPWM_setCounterCompareShadowLoadEvent
hrpwm.h	HRPWM_setOutputSwapMode
hrpwm.h	HRPWM_setChannelBOutputPath
hrpwm.h	HRPWM_enableAutoConversion
hrpwm.h	HRPWM_disableAutoConversion
hrpwm.h	HRPWM_setDeadbandMEPEdgeSelect
hrpwm.h	HRPWM_setRisingEdgeDelayLoadMode
hrpwm.h	HRPWM_setFallingEdgeDelayLoadMode
HRCNFG2	
hrpwm.h	HRPWM_setDeadbandMEPEdgeSelect
hrpwm.h	HRPWM_setRisingEdgeDelayLoadMode
hrpwm.h	HRPWM_setFallingEdgeDelayLoadMode
HRPCTL	
hrpwm.h	HRPWM_enablePeriodControl
hrpwm.h	HRPWM_disablePeriodControl
hrpwm.h	HRPWM_enablePhaseShiftLoad
hrpwm.h	HRPWM_disablePhaseShiftLoad
hrpwm.h	HRPWM_setSyncPulseSource
TRREM	
hrpwm.h	HRPWM_setTranslatorRemainder
GLDCTL	
epwm.h	EPWM_enableGlobalLoad
epwm.h	EPWM_disableGlobalLoad
epwm.h	EPWM_setGlobalLoadTrigger
epwm.h	EPWM_setGlobalLoadEventPrescale
epwm.h	EPWM_getGlobalLoadEventCount
epwm.h	EPWM_disableGlobalLoadOneShotMode
epwm.h	EPWM_enableGlobalLoadOneShotMode
epwm.h	EPWM_setGlobalLoadOneShotLatch
epwm.h	EPWM_forceGlobalLoadOneShotEvent
GLDCFG	
epwm.h	EPWM_enableGlobalLoadRegisters
epwm.h	EPWM_disableGlobalLoadRegisters
AQCTLA	
epwm.h	EPWM_setActionQualifierAction
epwm.h	EPWM_setActionQualifierActionComplete
epwm.h	EPWM_setAdditionalActionQualifierActionComplete
AQCTLA2	
epwm.h	EPWM_setActionQualifierAction
epwm.h	EPWM_setAdditionalActionQualifierActionComplete
AQCTLB	

Table 30-22. EPWM Registers to Driverlib Functions (continued)

File	Driverlib Function
-	See AQCTLA
AQCTLB2	
-	See AQCTLA2
AQSFRC	
epwm.h	EPWM_setActionQualifierContSWForceShadowMode
epwm.h	EPWM_setActionQualifierSWAction
epwm.h	EPWM_forceActionQualifierSWAction
AQCSFRC	
epwm.h	EPWM_setActionQualifierContSWForceAction
DBREDHR	
hrpwm.h	HRPWM_setRisingEdgeDelay
hrpwm.h	HRPWM_setHiResRisingEdgeDelayOnly
DBRED	
epwm.h	EPWM_setRisingEdgeDelayCount
hrpwm.h	HRPWM_setRisingEdgeDelay
hrpwm.h	HRPWM_setHiResRisingEdgeDelayOnly
DBFEDHR	
hrpwm.h	HRPWM_setFallingEdgeDelay
hrpwm.h	HRPWM_setHiResFallingEdgeDelayOnly
DBFED	
epwm.h	EPWM_setFallingEdgeDelayCount
hrpwm.h	HRPWM_setFallingEdgeDelay
hrpwm.h	HRPWM_setHiResFallingEdgeDelayOnly
TBPHS	
epwm.h	EPWM_setPhaseShift
hrpwm.h	HRPWM_setPhaseShift
hrpwm.h	HRPWM_setHiResPhaseShiftOnly
TBPRDHR	
hrpwm.h	HRPWM_setTimeBasePeriod
hrpwm.h	HRPWM_setHiResTimeBasePeriodOnly
hrpwm.h	HRPWM_getTimeBasePeriod
hrpwm.h	HRPWM_getHiResTimeBasePeriodOnly
TBPRD	
epwm.h	EPWM_setTimeBasePeriod
epwm.h	EPWM_getTimeBasePeriod
hrpwm.h	HRPWM_setTimeBasePeriod
hrpwm.h	HRPWM_setHiResTimeBasePeriodOnly
hrpwm.h	HRPWM_getTimeBasePeriod
hrpwm.h	HRPWM_getHiResTimeBasePeriodOnly
CMPA	
epwm.h	EPWM_setCounterCompareValue
epwm.h	EPWM_getCounterCompareValue
hrpwm.h	HRPWM_setCounterCompareValue
hrpwm.h	HRPWM_setHiResCounterCompareValueOnly
hrpwm.h	HRPWM_getCounterCompareValue

Table 30-22. EPWM Registers to Driverlib Functions (continued)

File	Driverlib Function
hrpwm.h	HRPWM_getHiResCounterCompareValueOnly
CMPB	
hrpwm.h	HRPWM_setCounterCompareValue
hrpwm.h	HRPWM_setHiResCounterCompareValueOnly
hrpwm.h	HRPWM_getCounterCompareValue
hrpwm.h	HRPWM_getHiResCounterCompareValueOnly
CMPC	
epwm.h	EPWM_setCounterCompareShadowLoadMode
epwm.h	EPWM_disableCounterCompareShadowLoadMode
epwm.h	EPWM_getCounterCompareShadowStatus
epwm.h	EPWM_enableLinkDutyHR
epwm.h	EPWM_disableLinkDutyHR
epwm.h	EPWM_setCMPShadowRegValue
CMPD	
-	See CMPC
GLDCTL2	
epwm.h	EPWM_setGlobalLoadOneShotLatch
epwm.h	EPWM_forceGlobalLoadOneShotEvent
SWVDELVAL	
epwm.h	EPWM_setValleySWDelayValue
TZSEL	
epwm.h	EPWM_enableTripZoneSignals
epwm.h	EPWM_disableTripZoneSignals
TZSEL2	
-	
TZDCSEL	
epwm.h	EPWM_setTripZoneDigitalCompareEventCondition
TZCTL	
epwm.h	EPWM_enableTripZoneAdvAction
epwm.h	EPWM_disableTripZoneAdvAction
epwm.h	EPWM_setTripZoneAction
epwm.h	EPWM_setTripZoneAdvAction
epwm.h	EPWM_setTripZoneAdvDigitalCompareActionA
epwm.h	EPWM_setTripZoneAdvDigitalCompareActionB
TZCTL2	
epwm.h	EPWM_enableTripZoneAdvAction
epwm.h	EPWM_disableTripZoneAdvAction
epwm.h	EPWM_setTripZoneAdvAction
epwm.h	EPWM_setTripZoneAdvDigitalCompareActionA
epwm.h	EPWM_setTripZoneAdvDigitalCompareActionB
TZCTLDCA	
epwm.h	EPWM_setTripZoneAdvDigitalCompareActionA
TZCTLDCB	
epwm.h	EPWM_setTripZoneAdvDigitalCompareActionB
TZEINT	

Table 30-22. EPWM Registers to Driverlib Functions (continued)

File	Driverlib Function
epwm.h	EPWM_enableTripZoneInterrupt
epwm.h	EPWM_disableTripZoneInterrupt
TZFLG	
epwm.h	EPWM_getTripZoneFlagStatus
TZCBCFLG	
epwm.h	EPWM_getCycleByCycleTripZoneFlagStatus
TZOSTFLG	
epwm.h	EPWM_getOneShotTripZoneFlagStatus
TZCLR	
epwm.h	EPWM_selectCycleByCycleTripZoneClearEvent
epwm.h	EPWM_clearTripZoneFlag
TZCBCCLR	
epwm.h	EPWM_clearCycleByCycleTripZoneFlag
TZOSTCLR	
epwm.h	EPWM_clearOneShotTripZoneFlag
TZFRC	
epwm.h	EPWM_forceTripZoneEvent
TZTRIPOUTSEL	
epwm.h	EPWM_enableTripOutSource
epwm.h	EPWM_disableTripOutSource
ETSEL	
epwm.h	EPWM_enableInterrupt
epwm.h	EPWM_disableInterrupt
epwm.h	EPWM_setInterruptSource
epwm.h	EPWM_enableADCTrigger
epwm.h	EPWM_disableADCTrigger
epwm.h	EPWM_setADCTriggerSource
ETPS	
epwm.h	EPWM_setInterruptEventCount
epwm.h	EPWM_setADCTriggerEventPrescale
ETFLG	
epwm.h	EPWM_getEventTriggerInterruptStatus
epwm.h	EPWM_getADCTriggerFlagStatus
ETCLR	
epwm.h	EPWM_clearEventTriggerInterruptFlag
epwm.h	EPWM_clearADCTriggerFlag
ETFRC	
epwm.h	EPWM_forceEventTriggerInterrupt
epwm.h	EPWM_forceADCTrigger
ETINTPS	
epwm.h	EPWM_setInterruptEventCount
epwm.h	EPWM_getInterruptEventCount
ETSOCPS	
epwm.h	EPWM_setADCTriggerEventPrescale
epwm.h	EPWM_getADCTriggerEventCount

Table 30-22. EPWM Registers to Driverlib Functions (continued)

File	Driverlib Function
ETCNTINITCTL	
epwm.h	EPWM_enableInterruptEventCountInit
epwm.h	EPWM_disableInterruptEventCountInit
epwm.h	EPWM_forceInterruptEventCountInit
epwm.h	EPWM_enableADCTriggerEventCountInit
epwm.h	EPWM_disableADCTriggerEventCountInit
epwm.h	EPWM_forceADCTriggerEventCountInit
ETCNTINIT	
epwm.h	EPWM_enableInterruptEventCountInit
epwm.h	EPWM_disableInterruptEventCountInit
epwm.h	EPWM_forceInterruptEventCountInit
epwm.h	EPWM_setInterruptEventCountInitValue
epwm.h	EPWM_enableADCTriggerEventCountInit
epwm.h	EPWM_disableADCTriggerEventCountInit
epwm.h	EPWM_forceADCTriggerEventCountInit
epwm.h	EPWM_setADCTriggerEventCountInitValue
ETINTMIXEN	
epwm.h	EPWM_setMixEvtTriggerSource
ETSOCAMIXEN	
-	
ETSOCBMIXEN	
-	
DCTRIPSEL	
epwm.h	EPWM_selectDigitalCompareTripInput
epwm.h	EPWM_enableDigitalCompareTripCombinationInput
DCACTL	
epwm.h	EPWM_setDigitalCompareEventSource
epwm.h	EPWM_setDigitalCompareEventSyncMode
epwm.h	EPWM_enableDigitalCompareADCTrigger
epwm.h	EPWM_disableDigitalCompareADCTrigger
epwm.h	EPWM_enableDigitalCompareSyncEvent
epwm.h	EPWM_disableDigitalCompareSyncEvent
epwm.h	EPWM_setDigitalCompareCBCLatchMode
epwm.h	EPWM_selectDigitalCompareCBCLatchClearEvent
epwm.h	EPWM_getDigitalCompareCBCLatchStatus
DCBCTL	
-	See DCACTL
DCFCTL	
epwm.h	EPWM_enableDigitalCompareBlankingWindow
epwm.h	EPWM_disableDigitalCompareBlankingWindow
epwm.h	EPWM_enableDigitalCompareWindowInverseMode
epwm.h	EPWM_disableDigitalCompareWindowInverseMode
epwm.h	EPWM_setDigitalCompareBlankingEvent
epwm.h	EPWM_setDigitalCompareFilterInput
epwm.h	EPWM_enableDigitalCompareEdgeFilter

Table 30-22. EPWM Registers to Driverlib Functions (continued)

File	Driverlib Function
epwm.h	EPWM_disableDigitalCompareEdgeFilter
epwm.h	EPWM_setDigitalCompareEdgeFilterMode
epwm.h	EPWM_setDigitalCompareEdgeFilterEdgeCount
epwm.h	EPWM_getDigitalCompareEdgeFilterEdgeCount
epwm.h	EPWM_getDigitalCompareEdgeFilterEdgeStatus
DCCAPCTL	
epwm.h	EPWM_enableDigitalCompareCounterCapture
epwm.h	EPWM_disableDigitalCompareCounterCapture
epwm.h	EPWM_setDigitalCompareCounterShadowMode
epwm.h	EPWM_getDigitalCompareCaptureStatus
epwm.h	EPWM_configureDigitalCompareCounterCaptureMode
epwm.h	EPWM_clearDigitalCompareCaptureStatusFlag
DCFOFFSET	
epwm.h	EPWM_setDigitalCompareWindowOffset
epwm.h	EPWM_getDigitalCompareBlankingWindowOffsetCount
DCFOFFSETCNT	
epwm.h	EPWM_getDigitalCompareBlankingWindowOffsetCount
DCFWINDOW	
epwm.h	EPWM_setDigitalCompareWindowLength
epwm.h	EPWM_getDigitalCompareBlankingWindowLengthCount
DCFWINDOWCNT	
epwm.h	EPWM_getDigitalCompareBlankingWindowLengthCount
BLANKPULSEMIXSEL	
-	
DCCAPMIXSEL	
-	
DCCAP	
epwm.h	EPWM_enableDigitalCompareCounterCapture
epwm.h	EPWM_disableDigitalCompareCounterCapture
epwm.h	EPWM_setDigitalCompareCounterShadowMode
epwm.h	EPWM_getDigitalCompareCaptureStatus
epwm.h	EPWM_configureDigitalCompareCounterCaptureMode
epwm.h	EPWM_clearDigitalCompareCaptureStatusFlag
epwm.h	EPWM_getDigitalCompareCaptureCount
DCAHTRIPSEL	
epwm.h	EPWM_enableDigitalCompareTripCombinationInput
epwm.h	EPWM_disableDigitalCompareTripCombinationInput
DCALTRIPSEL	
-	See DCAHTRIPSEL
DCBHTRIPSEL	
-	See DCAHTRIPSEL
DCBLTRIPSEL	
-	See DCAHTRIPSEL
CAPCTL	
epwm.h	EPWM_enableCaptureInEvent

Table 30-22. EPWM Registers to Driverlib Functions (continued)

File	Driverlib Function
epwm.h	EPWM_disableCaptureInEvent
epwm.h	EPWM_configCaptureGateInputPolarity
epwm.h	EPWM_invertCaptureInputPolarity
epwm.h	EPWM_enableIndependentPulseLogic
epwm.h	EPWM_disableIndependentPulseLogic
epwm.h	EPWM_forceCaptureEventLoad
CAPGATETRIPSEL	
epwm.h	EPWM_enableCaptureTripCombinationInput
epwm.h	EPWM_disableCaptureTripCombinationInput
CAPINTRIPSEL	
-	
CAPTRIPSEL	
epwm.h	EPWM_selectCaptureTripInput
epwm.h	EPWM_enableCaptureTripCombinationInput
LOCK	
epwm.h	EPWM_lockRegisters
hrpwm.h	HRPWM_lockRegisters
HWVDELVAL	
epwm.h	EPWM_getValleyHWDelay
VCNTVAL	
epwm.h	EPWM_getValleyCount
DECTL	
epwm.h	EPWM_enableDiodeEmulationMode
epwm.h	EPWM_disableDiodeEmulationMode
epwm.h	EPWM_setDiodeEmulationMode
epwm.h	EPWM_setDiodeEmulationReentryDelay
DECOMPSEL	
epwm.h	EPWM_configureDiodeEmulationTripLowSources
epwm.h	EPWM_configureDiodeEmulationTripHighSources
DEACTCTL	
epwm.h	EPWM_selectDiodeEmulationPWMsignal
epwm.h	EPWM_selectDiodeEmulationTripSignal
epwm.h	EPWM_nobypassDiodeEmulationLogic
epwm.h	EPWM_bypassDiodeEmulationLogic
DESTS	
-	
DEFRC	
epwm.h	EPWM_forceDiodeEmulationActive
DECLR	
epwm.h	EPWM_clearDiodeEmulationActiveFlag
DEMONCNT	
-	
DEMONCTL	
epwm.h	EPWM_enableDiodeEmulationMonitorModeControl
epwm.h	EPWM_disableDiodeEmulationMonitorModeControl

Table 30-22. EPWM Registers to Driverlib Functions (continued)

File	Driverlib Function
DEMONSTEP	
epwm.h	EPWM_setDiodeEmulationMonitorModeStep
DEMONTHRES	
epwm.h	EPWM_setDiodeEmulationMonitorCounterThreshold
MINDBCFG	
epwm.h	EPWM_enableMinimumDeadBand
epwm.h	EPWM_invertMinimumDeadBandSignal
epwm.h	EPWM_selectMinimumDeadBandAndOrLogic
epwm.h	EPWM_selectMinimumDeadBandBlockingSignal
epwm.h	EPWM_selectMinimumDeadBandReferenceSignal
MINDBDLY	
epwm.h	EPWM_getMinDeadBandDelay
epwm.h	EPWM_setMinDeadBandDelay
LUTCTLA	
epwm.h	EPWM_enableIllegalComboLogic
epwm.h	EPWM_disableIllegalComboLogic
epwm.h	EPWM_selectXbarInput
epwm.h	EPWM_setLutDecX
LUTCTLB	
epwm.h	EPWM_enableIllegalComboLogic
epwm.h	EPWM_disableIllegalComboLogic
epwm.h	EPWM_selectXbarInput
epwm.h	EPWM_setLutDecX

30.19.2 HRPWMCAL Registers to Driverlib Functions

Table 30-23. HRPWMCAL Registers to Driverlib Functions

File	Driverlib Function
HRPWR	
-	
HRMSTEP	
hrpwm.h	HRPWM_setMEPStep

30.19.3 EPWM Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/epwm

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

30.19.3.1 ePWM Trip Zone - SINGLE_CORE

FILE: epwm_ex1_trip_zone.c

This example configures ePWM1 and ePWM2 as follows

- ePWM1 has TZ1 as one shot trip source
- ePWM2 has TZ1 as cycle by cycle trip source

Initially tie TZ1 high. During the test, monitor ePWM1 or ePWM2 outputs on a scope. Pull TZ1 low to see the effect.

External Connections

- ePWM1A is on GPIO0
- ePWM2A is on GPIO2
- TZ1 is on GPIO12

This example also makes use of the Input X-BAR. GPIO12 (the external trigger) is routed to the input X-BAR, from which it is routed to TZ1.

The TZ-Event is defined such that ePWM1A will undergo a One-Shot Trip and ePWM2A will undergo a Cycle-By-Cycle Trip.

30.19.3.2 ePWM Up Down Count Action Qualifier - SINGLE_CORE

FILE: epwm_ex2_updown_aq.c

This example configures ePWM1, ePWM2, ePWM3 to produce a waveform with independent modulation on ePWMxA and ePWMxB.

The compare values CMPA and CMPB are modified within the ePWM's ISR.

The TB counter is in up/down count mode for this example.

View the ePWM1A/B(GPIO0 & GPIO1), ePWM2A/B(GPIO2 &GPIO3) and ePWM3A/B(GPIO4 & GPIO5) waveforms on oscilloscope.

30.19.3.3 ePWM Synchronization - SINGLE_CORE

FILE: epwm_ex3_synchronization.c

This example configures ePWM1, ePWM2, ePWM3 and ePWM4 as follows

- ePWM1 without phase shift as sync source
- ePWM2 with phase shift of 300 TBCLKs
- ePWM3 with phase shift of 600 TBCLKs
- ePWM4 with phase shift of 900 TBCLKs

External Connections

- GPIO0 EPWM1A
- GPIO1 EPWM1B
- GPIO2 EPWM2A
- GPIO3 EPWM2B
- GPIO4 EPWM3A
- GPIO5 EPWM3B
- GPIO6 EPWM4A
- GPIO7 EPWM4B

Watch Variables

- None.

30.19.3.4 ePWM Digital Compare - SINGLE_CORE

FILE: epwm_ex4_digital_compare.c

This example configures ePWM1 as follows

- ePWM1 with DCAEVT1 forcing the ePWM output LOW
- GPIO25 is used as the input to the INPUT XBAR INPUT1
- INPUT1 (from INPUT XBAR) is used as the source for DCAEVT1
- GPIO25's PULL-UP resistor is enabled, in order to test the trip, PULL this pin to GND

External Connections

- GPIO0 EPWM1A
- GPIO1 EPWM1B

- GPIO25 TZ1, pull this pin low to trip the ePWM

Watch Variables

- None.

30.19.3.5 ePWM Digital Compare Event Filter Blanking Window - SINGLE_CORE

FILE: epwm_ex5_digital_compare_event_filter.c

This example configures ePWM1 as follows

- ePWM1 with DCAEVT1 forcing the ePWM output LOW
- GPIO25 is used as the input to the INPUT XBAR INPUT1
- INPUT1 (from INPUT XBAR) is used as the source for DCAEVT1
- GPIO25's PULL-UP resistor is enabled, in order to test the trip, PULL this pin to GND
- ePWM1 with DCBEVT1 forcing the ePWM output LOW
- GPIO25 is used as the input to the INPUT XBAR INPUT1
- INPUT1 (from INPUT XBAR) is used as the source for DCAEVT1
- GPIO25's PULL-UP resistor is enabled, in order to test the trip, PULL this pin to GND
- DCBEVT1 uses the filtered version of DCBEVT1
- The DCFILT signal uses the blanking window to ignore the DCBEVT1 for the duration of DC Blanking window

External Connections

- GPIO0 EPWM1A
- GPIO1 EPWM1B
- GPIO25 TRIPIN1, pull this pin low to trip the ePWM

Watch Variables

- None.

30.19.3.6 ePWM Valley Switching - SINGLE_CORE

FILE: epwm_ex6_valley_switching.c

This example configures ePWM1 as follows

- ePWM1 with DCAEVT1 forcing the ePWM output LOW
- GPIO25 is used as the input to the INPUT XBAR INPUT1
- INPUT1 (from INPUT XBAR) is used as the source for DCAEVT1
- GPIO25 is set to output and toggled in the main loop to trip the PWM
- ePWM1 with DCBEVT1 forcing the ePWM output LOW
- GPIO25 is used as the input to the INPUT XBAR INPUT1
- INPUT1 (from INPUT XBAR) is used as the source for DCAEVT1
- GPIO25 is set to output and toggled in the main loop to trip the PWM
- DCBEVT1 uses the filtered version of DCBEVT1
- The DCFILT signal uses the valley switching module to delay the
- DCFILT signal by a software defined DELAY value.

External Connections

- GPIO0 EPWM1A
- GPIO1 EPWM1B
- GPIO25 TRIPIN1 (Output Pin, toggled through software)

Watch Variables

- None.

30.19.3.7 ePWM Digital Compare Edge Filter - SINGLE_CORE

FILE: epwm_ex7_edge_filter.c

This example configures ePWM1 as follows

- ePWM1 with DCBEVT2 forcing the ePWM output LOW as a CBC source
- GPIO25 is used as the input to the INPUT XBAR INPUT1
- INPUT1 (from INPUT XBAR) is used as the source for DCBEVT2
- GPIO25 is set to output and toggled in the main loop to trip the PWM
- The DCBEVT2 is the source for DCFILT
- The DCFILT will count edges of the DCBEVT2 and generate a signal to to trip the ePWM on the 4th edge of DCBEVT2

External Connections

- GPIO0 EPWM1A
- GPIO1 EPWM1B
- GPIO25 TRIPIN1 (Output Pin, toggled through software)

Watch Variables

- None.

30.19.3.8 ePWM Deadband - SINGLE_CORE

FILE: epwm_ex8_deadband.c

This example configures ePWM1 through ePWM6 as follows

- ePWM1 with Deadband disabled (Reference)
- ePWM2 with Deadband Active High
- ePWM3 with Deadband Active Low
- ePWM4 with Deadband Active High Complimentary
- ePWM5 with Deadband Active Low Complimentary
- ePWM6 with Deadband Output Swap (switch A and B outputs)

External Connections

- GPIO0 EPWM1A
- GPIO1 EPWM1B
- GPIO2 EPWM2A
- GPIO3 EPWM2B
- GPIO4 EPWM3A
- GPIO5 EPWM3B
- GPIO6 EPWM4A
- GPIO7 EPWM4B
- GPIO8 EPWM5A
- GPIO9 EPWM5B
- GPIO14 EPWM6A
- GPIO11 EPWM6B

Watch Variables

- None.

30.19.3.9 ePWM DMA - SINGLE_CORE

FILE: epwm_ex9_dma.c

This example configures ePWM1 and DMA as follows:

- ePWM1 is set up to generate PWM waveforms
- DMA5 is set up to update the CMPAHR, CMPA, CMPBHR and CMPB every period with the next value in the configuration array. This allows the user to create a DMA enabled fifo for all the CMPx and CMPxHR registers to generate unconventional PWM waveforms.
- DMA6 is set up to update the TBPHSHR, TBPHS, TBPRDHR and TBPRD every period with the next value in the configuration array.
- Other registers such as AQCTL can be controlled through the DMA as well by following the same procedure. (Not used in this example)

External Connections
- GPIO0 EPWM1A
- GPIO1 EPWM1B

Watch Variables
- None.

30.19.3.10 ePWM Chopper - SINGLE_CORE

FILE: epwm_ex10_chopper.c

This example configures ePWM1, ePWM2, ePWM3 and ePWM4 as follows
- ePWM1 with Chopper disabled (Reference)
- ePWM2 with chopper enabled at 1/8 duty cycle
- ePWM3 with chopper enabled at 6/8 duty cycle
- ePWM4 with chopper enabled at 1/2 duty cycle with One-Shot Pulse enabled

External Connections
- GPIO0 EPWM1A
- GPIO1 EPWM1B
- GPIO2 EPWM2A
- GPIO3 EPWM2B
- GPIO4 EPWM3A
- GPIO5 EPWM3B
- GPIO6 EPWM4A
- GPIO7 EPWM4B

Watch Variables
- None.

30.19.3.11 EPWM Configure Signal - SINGLE_CORE

FILE: epwm_ex11_configure_signal.c

This example configures ePWM1, ePWM2, ePWM3 to produce signal of desired frequency and duty. It also configures phase between the configured modules.

Signal of 10kHz with duty of 0.5 is configured on ePWMxA & ePWMxB with ePWMxB inverted. Also, phase of 120 degree is configured between ePWM1 to ePWM3 signals.

During the test, monitor ePWM1, ePWM2, and/or ePWM3 outputs on an oscilloscope.

- ePWM1A is on GPIO0
- ePWM1B is on GPIO1
- ePWM2A is on GPIO2
- ePWM2B is on GPIO3
- ePWM3A is on GPIO4
- ePWM3B is on GPIO5

30.19.3.12 Realization of Monoshot mode - SINGLE_CORE

FILE: epwm_ex12_monoshot_mode.c

This example showcases how to generate monoshot PWM output based on external trigger i.e. generating just a single pulse output on receipt of an external trigger. And the next pulse will be generated only when the next trigger comes. The example utilizes external synchronization and T1 action qualifier event features to achieve the desired output.

ePWM1 is used to generate the monoshot output and ePWM2 is used an external trigger for that. No external connections are required as ePWM2A is fed as the trigger using Input X-BAR automatically.

ePWM1 is configured to generated a single pulse of 0.5us when received an external trigger. This is achieved by enabling the phase synchronization feature and configuring EPWMxSYNCI as EXTSYNCIN1. And this EPWMxSYNCI is also configured as T1 event of action qualifier to set output HIGH while "CTR = PRD" action is used to set output LOW.

ePWM2 is configured to generate a 100 KHz signal with a duty of 1% (to simulate a rising edge trigger) which is routed to EXTSYNCIN1 using Input XBAR.

Observe GPIO0 (EPWM1A : Monoshot Output) and GPIO2(EPWM2 : External Trigger) on oscilloscope.

30.19.3.13 EPWM Action Qualifier (epwm_up_aq) - SINGLE_CORE

FILE: epwm_ex13_up_aq.c

This example configures ePWM1, ePWM2, ePWM3 to produce an waveform with independent modulation on EPWMxA and EPWMxB.

The compare values CMPA and CMPB are modified within the ePWM's ISR.

The TB counter is in up count mode for this example.

View the EPWM1A/B(GPIO0 & GPIO1), EPWM2A/B(GPIO2 & GPIO3) and EPWM3A/B(GPIO4 & GPIO5) waveforms via an oscilloscope.

30.19.3.14 ePWM XCMP Mode - SINGLE_CORE

FILE: epwm_ex15_xcmp_multiple_edges.c

(Note - base frequency and duty cycle of all ePWM's are 50 KHz and 50% respectively. Value of TBPRD = 1999)
This example configures ePWM1, ePWM2, ePWM4, ePWM6 and ePWM8 as follows

- ePWM1A is allocated all XCMP1-8 registers. ePWM1B has no output.
  - New duty cycle = 50%, new frequency = 200 KHz
  - No Shadow registers used
- ePWM2A is allocated XCMP1-4 and ePWM2B is allocated XCMP5-8 registers.
  - A and B waveforms are complimentary
  - New duty cycle = 50%, new frequency = 100 KHz
  - No Shadow registers used
- ePWM4 is configured same as ePWM2 with Minimum Deadband.
  - Minimum Deadband of 200 SYSCLK cycles provided (200 * (1/200 MHz) = 1 micro second)
  - This implies dead band of 1 us is visible on output of ePWM4A and ePWM4B after their falling edge
  - New duty cycle = 40%, new frequency = 100 KHz
- ePWM6A is allocated XCMP1-4 registers.
  - 3 Shadow register sets used with LOADMULTIPLE mode
  - Shadow set 2 repeated 2 times, Shadow set 3 repeated 4 times
  - ISR to update all Shadow registers with new values after they repeat
  - This means Shadow3 is active for 5 periods, Shadow2 is active for 3 periods and Shadow1 is active for 1 period before their new values are visible in output
- ePWM8A is allocated XCMP1-4 registers.
  - 3 Shadow register sets used with LOADONCE mode
  - Only Shadow set 3 is loaded from every period
  - ISR updates Shadow 3 register with new values every 5 periods

External Connections

- ePWM1A is on GPIO0
- ePWM2A is on GPIO2 and ePWM2B is on GPIO3
- ePWM4A is on GPIO6 and ePWM4B is on GPIO7
- ePWM6A is on GPIO14
- ePWM8A is on GPIO10

- Monitor GPIO24 for ePWM6A new Shadow register value loading
- Monitor GPIO25 for ePWM8A new Shadow register value loading

Shadow register updations for ePWM6 and ePWM8:
- Only XCMP1 and XCMP4 are updated
- Update values are +/- 20 TBCTR steps depending on direction of updation
- GPIO24 is toggled every 9 cycles for ePWM6 after cycling through all Shadow buffers and loading new values Similarly GPIO25 is toggled every 5 cycles for ePWM8

30.19.3.15 ePWM Event Detection - SINGLE_CORE

FILE: epwm_ex16_event_detection.c

(Note - base frequency and duty cycle of all ePWM's are 50 KHz and 50% respectively. Value of TBPRD = 1999) This example configures ePWM1 and ePWM2 in identical fashion with XCMP1-8 allocated to channel A. No shadow registers are used.
- XCMP1 = 250, XMP2 = 500, XCMP3 = 750 ..... XCMP 8 = 1750.
- In ePWM1, XMIN = 300 and XMAX = 400.
  - This window has no edge and generates a CAPEVT pulse every period.
- In ePWM2, XMIN = 300 and XMAX = 600.
  - This window has an edge and doesn't generate CAPEVT pulse signal.

External Connections
- ePWM1A is on GPIO0
- ePWM2A is on GPIO2
- ePWM1 Tripout is on GPIO24
- ePWM2 Tripout is on GPIO25
- LED1 is on GPIO31 (For control card)
- LED2 is on GPIO34 (For control card)

CAPIN and CAPGATE signals are both sourced as Trip4 for ePWM1 and Trip5 for ePWM2. Trip4 and Trip5 are routed from the INPUT X-BAR through EPWM-XBAR which feeds into the Digital Compare submodule. CAPEVT signal is used as Tripout and Trip-Zone interrupt source.

For ePWM1, Trip-Zone ISR configured to make LED1 blink 1 second on/off. For ePWM2, Trip-Zone ISR configured to make LED2 turn on if an interrupt ever occurs.

30.20 EPWM Registers

This Section describes the EPWM Registers.

30.20.1 EPWM Base Address Table

Table 30-24. EPWM Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
EPWM_REGS	EPWM1_BASE	0x7000_0000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM1XCMP_BASE	0x7000_0400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM1DE_BASE	0x7000_0800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM1MINDBLUT_BASE	0x7000_0C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM2_BASE	0x7000_1000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM2XCMP_BASE	0x7000_1400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM2DE_BASE	0x7000_1800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM2MINDBLUT_BASE	0x7000_1C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM3_BASE	0x7000_2000	YES	YES	YES	YES	YES	YES	-	YES

Table 30-24. EPWM Base Address Table (continued)

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
EPWM_XCMP_REGS	EPWM3XCMP_BASE	0x7000_2400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM3DE_BASE	0x7000_2800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM3MINDBLUT_BASE	0x7000_2C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM4_BASE	0x7000_3000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM4XCMP_BASE	0x7000_3400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM4DE_BASE	0x7000_3800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM4MINDBLUT_BASE	0x7000_3C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM5_BASE	0x7000_4000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM5XCMP_BASE	0x7000_4400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM5DE_BASE	0x7000_4800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM5MINDBLUT_BASE	0x7000_4C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM6_BASE	0x7000_5000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM6XCMP_BASE	0x7000_5400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM6DE_BASE	0x7000_5800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM6MINDBLUT_BASE	0x7000_5C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM7_BASE	0x7000_6000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM7XCMP_BASE	0x7000_6400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM7DE_BASE	0x7000_6800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM7MINDBLUT_BASE	0x7000_6C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM8_BASE	0x7000_7000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM8XCMP_BASE	0x7000_7400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM8DE_BASE	0x7000_7800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM8MINDBLUT_BASE	0x7000_7C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM9_BASE	0x7000_8000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM9XCMP_BASE	0x7000_8400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM9DE_BASE	0x7000_8800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM9MINDBLUT_BASE	0x7000_8C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM10_BASE	0x7000_9000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM10XCMP_BASE	0x7000_9400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM10DE_BASE	0x7000_9800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM10MINDBLUT_BASE	0x7000_9C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM11_BASE	0x7000_A000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM11XCMP_BASE	0x7000_A400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM11DE_BASE	0x7000_A800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM11MINDBLUT_BASE	0x7000_AC00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM12_BASE	0x7000_B000	YES	YES	YES	YES	YES	YES	-	YES

Table 30-24. EPWM Base Address Table (continued)

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
EPWM_XCMP_REGS	EPWM12XCMP_BASE	0x7000_B400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM12DE_BASE	0x7000_B800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM12MINDBLUT_BASE	0x7000_BC00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM13_BASE	0x7000_C000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM13XCMP_BASE	0x7000_C400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM13DE_BASE	0x7000_C800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM13MINDBLUT_BASE	0x7000_CC00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM14_BASE	0x7000_D000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM14XCMP_BASE	0x7000_D400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM14DE_BASE	0x7000_D800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM14MINDBLUT_BASE	0x7000_DC00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM15_BASE	0x7000_E000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM15XCMP_BASE	0x7000_E400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM15DE_BASE	0x7000_E800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM15MINDBLUT_BASE	0x7000_EC00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM16_BASE	0x7000_F000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM16XCMP_BASE	0x7000_F400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM16DE_BASE	0x7000_F800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM16MINDBLUT_BASE	0x7000_FC00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM17_BASE	0x7001_0000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM17XCMP_BASE	0x7001_0400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM17DE_BASE	0x7001_0800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM17MINDBLUT_BASE	0x7001_0C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM18_BASE	0x7001_1000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM18XCMP_BASE	0x7001_1400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM18DE_BASE	0x7001_1800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM18MINDBLUT_BASE	0x7001_1C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM1XLINK_BASE	0x7004_0000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM1XCMPXLINK_BASE	0x7004_0400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM1DEXLINK_BASE	0x7004_0800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM1MINDBLUTXLINK_BASE	0x7004_0C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM2XLINK_BASE	0x7004_1000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM2XCMPXLINK_BASE	0x7004_1400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM2DEXLINK_BASE	0x7004_1800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM2MINDBLUTXLINK_BASE	0x7004_1C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM3XLINK_BASE	0x7004_2000	YES	YES	YES	YES	YES	YES	-	YES

Table 30-24. EPWM Base Address Table (continued)

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
EPWM_XCMP_REGS	EPWM3XCMPXLINK_BASE	0x7004_2400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM3DEXLINK_BASE	0x7004_2800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM3MINDBLUTXLINK_BASE	0x7004_2C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM4XLINK_BASE	0x7004_3000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM4XCMPXLINK_BASE	0x7004_3400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM4DEXLINK_BASE	0x7004_3800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM4MINDBLUTXLINK_BASE	0x7004_3C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM5XLINK_BASE	0x7004_4000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM5XCMPXLINK_BASE	0x7004_4400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM5DEXLINK_BASE	0x7004_4800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM5MINDBLUTXLINK_BASE	0x7004_4C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM6XLINK_BASE	0x7004_5000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM6XCMPXLINK_BASE	0x7004_5400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM6DEXLINK_BASE	0x7004_5800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM6MINDBLUTXLINK_BASE	0x7004_5C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM7XLINK_BASE	0x7004_6000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM7XCMPXLINK_BASE	0x7004_6400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM7DEXLINK_BASE	0x7004_6800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM7MINDBLUTXLINK_BASE	0x7004_6C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM8XLINK_BASE	0x7004_7000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM8XCMPXLINK_BASE	0x7004_7400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM8DEXLINK_BASE	0x7004_7800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM8MINDBLUTXLINK_BASE	0x7004_7C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM9XLINK_BASE	0x7004_8000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM9XCMPXLINK_BASE	0x7004_8400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM9DEXLINK_BASE	0x7004_8800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM9MINDBLUTXLINK_BASE	0x7004_8C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM10XLINK_BASE	0x7004_9000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM10XCMPXLINK_BASE	0x7004_9400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM10DEXLINK_BASE	0x7004_9800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM10MINDBLUTXLINK_BASE	0x7004_9C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM11XLINK_BASE	0x7004_A000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM11XCMPXLINK_BASE	0x7004_A400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM11DEXLINK_BASE	0x7004_A800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM11MINDBLUTXLINK_BASE	0x7004_AC00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM12XLINK_BASE	0x7004_B000	YES	YES	YES	YES	YES	YES	-	YES

Table 30-24. EPWM Base Address Table (continued)

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
EPWM_XCMP_REGS	EPWM12XCMPXLINK_BASE	0x7004_B400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM12DEXLINK_BASE	0x7004_B800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM12MINDBLUTXLINK_BASE	0x7004_BC00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM13XLINK_BASE	0x7004_C000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM13XCMPXLINK_BASE	0x7004_C400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM13DEXLINK_BASE	0x7004_C800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM13MINDBLUTXLINK_BASE	0x7004_CC00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM14XLINK_BASE	0x7004_D000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM14XCMPXLINK_BASE	0x7004_D400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM14DEXLINK_BASE	0x7004_D800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM14MINDBLUTXLINK_BASE	0x7004_DC00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM15XLINK_BASE	0x7004_E000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM15XCMPXLINK_BASE	0x7004_E400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM15DEXLINK_BASE	0x7004_E800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM15MINDBLUTXLINK_BASE	0x7004_EC00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM16XLINK_BASE	0x7004_F000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM16XCMPXLINK_BASE	0x7004_F400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM16DEXLINK_BASE	0x7004_F800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM16MINDBLUTXLINK_BASE	0x7004_FC00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM17XLINK_BASE	0x7005_0000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM17XCMPXLINK_BASE	0x7005_0400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM17DEXLINK_BASE	0x7005_0800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM17MINDBLUTXLINK_BASE	0x7005_0C00	YES	YES	YES	YES	YES	YES	-	YES
EPWM_REGS	EPWM18XLINK_BASE	0x7005_1000	YES	YES	YES	YES	YES	YES	-	YES
EPWM_XCMP_REGS	EPWM18XCMPXLINK_BASE	0x7005_1400	YES	YES	YES	YES	YES	YES	-	YES
DE_REGS	EPWM18DEXLINK_BASE	0x7005_1800	YES	YES	YES	YES	YES	YES	-	YES
MINDB_LUT_REGS	EPWM18MINDBLUTXLINK_BASE	0x7005_1C00	YES	YES	YES	YES	YES	YES	-	YES
HRPWMCAL_REGS	HRPWMCAL1_BASE	0x7008_0000	YES	YES	YES	YES	YES	YES	-	YES
HRPWMCAL_REGS	HRPWMCAL2_BASE	0x7008_1000	YES	YES	YES	YES	YES	YES	-	YES
HRPWMCAL_REGS	HRPWMCAL3_BASE	0x7008_2000	YES	YES	YES	YES	YES	YES	-	YES

30.20.2 EPWM_REGS Registers

Table 30-25 lists the memory-mapped registers for the EPWM_REGS registers. All register offset addresses not listed in Table 30-25 should be considered as reserved locations and the register contents should not be modified.

Table 30-25. EPWM_REGS Registers

Offset	Acronym	Register Name	Protection
0h	TBCTL	Time Base Control Register	
2h	TBCTL2	Time Base Control Register 2	
6h	EPWMSYNCINSEL	EPWMxSYNCIN Source Select Register	
8h	TBCTR	Time Base Counter Register	
Ah	TBSTS	Time Base Status Register	
Ch	EPWMSYNCOUTEN	EPWMxSYNCOUT Source Enable Register	
Eh	TBCTL3	Time Base Control Register 3	
10h	CMPCTL	Counter Compare Control Register	
12h	CMPCTL2	Counter Compare Control Register 2	
18h	DBCTL	Dead-Band Generator Control Register	
1Ah	DBCTL2	Dead-Band Generator Control Register 2	
20h	AQCTL	Action Qualifier Control Register	
22h	AQTSRCSEL	Action Qualifier Trigger Event Source Select Register	
28h	PCCTL	PWM Chopper Control Register	
30h	VCAPCTL	Valley Capture Control Register	
32h	VCNTCFG	Valley Counter Config Register	
40h	HRCNFG	HRPWM Configuration Register	
4Eh	HRCNFG2	HRPWM Configuration 2 Register	
5Ah	HRPCTL	High Resolution Period Control Register	
5Ch	TRREM	HRPWM High Resolution Remainder Register	
68h	GLDCTL	Global PWM Load Control Register	
6Ah	GLDCFG	Global PWM Load Config Register	
80h	AQCTLA	Action Qualifier Control Register For Output A	
82h	AQCTLA2	Additional Action Qualifier Control Register For Output A	
84h	AQCTLB	Action Qualifier Control Register For Output B	
86h	AQCTLB2	Additional Action Qualifier Control Register For Output B	
8Eh	AQSFRC	Action Qualifier Software Force Register	
92h	AQCSFRC	Action Qualifier Continuous S/W Force Register	
A0h	DBREDHR	Dead-Band Generator Rising Edge Delay High Resolution Mirror Register	
A2h	DBRED	Dead-Band Generator Rising Edge Delay High Resolution Mirror Register	
A4h	DBFEDHR	Dead-Band Generator Falling Edge Delay High Resolution Register	
A6h	DBFED	Dead-Band Generator Falling Edge Delay Count Register	
C0h	TBPHS	Time Base Phase High	
C4h	TBPRDHR	Time Base Period High Resolution Register	
C6h	TBPRD	Time Base Period Register	
D4h	CMPA	Counter Compare A Register	
D8h	CMPB	Compare B Register	
DEh	CMPC	Counter Compare C Register	

Table 30-25. EPWM_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
E2h	CMPD	Counter Compare D Register	
E8h	GLDCTL2	Global PWM Load Control Register 2	
EEh	SWVDELVAL	Software Valley Mode Delay Register	
100h	TZSEL	Trip Zone Select Register	
102h	TZSEL2	Trip Zone Select Register 2	
104h	TZDCSEL	Trip Zone Digital Comparator Select Register	
108h	TZCTL	Trip Zone Control Register	
10Ah	TZCTL2	Additional Trip Zone Control Register	
10Ch	TZCTLDCA	Trip Zone Control Register Digital Compare A	
10Eh	TZCTLDCB	Trip Zone Control Register Digital Compare B	
11Ah	TZEINT	Trip Zone Enable Interrupt Register	
126h	TZFLG	Trip Zone Flag Register	
128h	TZCBCFLG	Trip Zone CBC Flag Register	
12Ah	TZOSTFLG	Trip Zone OST Flag Register	
12Eh	TZCLR	Trip Zone Clear Register	
130h	TZCBCCLR	Trip Zone CBC Clear Register	
132h	TZOSTCLR	Trip Zone OST Clear Register	
136h	TZFRC	Trip Zone Force Register	
13Ah	TZTRIPOUTSEL	Trip Zone Force Register	
148h	ETSEL	Event Trigger Selection Register	
14Ch	ETPS	Event Trigger Pre-Scale Register	
150h	ETFLG	Event Trigger Flag Register	
154h	ETCLR	Event Trigger Clear Register	
158h	ETFRC	Event Trigger Force Register	
15Ch	ETINTPS	Event-Trigger Interrupt Pre-Scale Register	
160h	ETSOCPS	Event-Trigger SOC Pre-Scale Register	
164h	ETCNTINITCTL	Event-Trigger Counter Initialization Control Register	
168h	ETCNTINIT	Event-Trigger Counter Initialization Register	
16Ch	ETINTMIXEN	Event-Trigger Mixed INT Selection	
170h	ETSOCAMIXEN	Event-Trigger Mixed SOCA Selection	
174h	ETSOCBMIXEN	Event-Trigger Mixed SOCB Selection	
180h	DCTRIPSEL	Digital Compare Trip Select Register	
186h	DCACTL	Digital Compare A Control Register	
188h	DCBCTL	Digital Compare B Control Register	
18Eh	DCFCTL	Digital Compare Filter Control Register	
190h	DCCAPCTL	Digital Compare Capture Control Register	
192h	DCFOFFSET	Digital Compare Filter Offset Register	
194h	DCFOFFSETCNT	Digital Compare Filter Offset Counter Register	
196h	DCFWINDOW	Digital Compare Filter Window Register	
198h	DCFWINDOWCNT	Digital Compare Filter Window Counter Register	
19Ah	BLANKPULSEMIXSEL	Blanking window trigger pulse select register	
19Ch	DCCAPMIXSEL	Capture Event pulse select register	
19Eh	DCCAP	Digital Compare Counter Capture Register	
1A4h	DCAHTRIPSEL	Digital Compare AH Trip Select	
1A6h	DCALTRIPSEL	Digital Compare AL Trip Select	

Table 30-25. EPWM_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
1A8h	DCBHTRIPSEL	Digital Compare BH Trip Select	
1AAh	DCBLTRIPSEL	Digital Compare BL Trip Select	
1ACh	CAPCTL	Event Capture Control Register	
1AEh	CAPGATETRIPSEL	Event Capture Gate Trip input select	
1B0h	CAPINTRIPSEL	Event Capture Trip input select	
1B2h	CAPTRIPSEL	Event Capture Signal Select	
1F4h	EPWMLOCK	EPWM Lock Register	
1FAh	HWVDELVAL	Hardware Valley Mode Delay Register	
1FCh	VCNTVAL	Hardware Valley Counter Register	

Complex bit access types are encoded to fit into small table cells. Table 30-26 shows the codes that are used for access types in this section.

Table 30-26. EPWM_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
WOnce	W Once	Write Write once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

30.20.2.1 TBCTL Register (Offset = 0h) [Reset = 0083h]

TBCTL is shown in Figure 30-122 and described in Table 30-27.

Return to the Summary Table.

Time Base Control Register

Figure 30-122. TBCTL Register

15	14	13	12	11	10	9	8
FREE_SOFT		PHSDIR	CLKDIV			HSPCLKDIV	
R/W-0h		R/W-0h	R/W-0h			R/W-1h	
7	6	5	4	3	2	1	0
HSPCLKDIV	SWFSYNC	RESERVED		PRDLD	PHSEN	CTRMODE	
R/W-1h	R-0/W1S-0h	R-0h		R/W-0h	R/W-0h	R/W-3h	

Table 30-27. TBCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	FREE_SOFT	R/W	0h	Emulation Mode Bits. These bits select the behavior of the ePWM time-base counter during emulation events 00: Stop after the next time-base counter increment or decrement 01: Stop when counter completes a whole cycle: - Up-count mode: stop when the time-base counter = period (TBCTR = TBPRD) - Down-count mode: stop when the time-base counter = 0x00 (TBCTR = 0x00) - Up-down-count mode: stop when the time-base counter = 0x00 (TBCTR = 0x00) 1x: Free run Reset type: SYSRSn
13	PHSDIR	R/W	0h	Phase Direction Bit This bit is only used when the time-base counter is configured in the up-down-count mode. The PHSDIR bit indicates the direction the time-base counter (TBCTR) will count after a synchronization event occurs and a new phase value is loaded from the phase (TBPHS) register. This is irrespective of the direction of the counter before the synchronization event.. In the up-count and down-count modes this bit is ignored. 0: Count down after the synchronization event. 1: Count up after the synchronization event. Reset type: SYSRSn
12-10	CLKDIV	R/W	0h	Time Base Clock Pre-Scale Bits These bits select the time base clock pre-scale value (TBCLK = EPWMCLK/(HSPCLKDIV * CLKDIV): 000: /1 (default on reset) 001: /2 010: /4 011: /8 100: /16 101: /32 110: /64 111: /128 Reset type: SYSRSn

Table 30-27. TBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9-7	HSPCLKDIV	R/W	1h	High Speed Time Base Clock Pre-Scale Bits These bits determine part of the time-base clock prescale value. TBCLK = EPWMCLK / (HSPCLKDIV x CLKDIV). This divisor emulates the HSPCLK in the TMS320x281x system as used on the Event Manager (EV) peripheral. 000: /1 001: /2 (default on reset) 010: /4 011: /6 100: /8 101: /10 110: /12 111: /14 Reset type: SYSRSn
6	SWFSYNC	R-0/W1S	0h	Software Forced Sync Pulse 0: Writing a 0 has no effect and reads always return a 0. 1: Writing a 1 forces a one-time synchronization pulse to be generated. SWFSYNC can be enabled to affect EPWMxSYNCO by setting the EPWMSYNCOUTEN.SWEN bit. Reset type: SYSRSn
5-4	RESERVED	R	0h	Reserved
3	PRDLD	R/W	0h	Active Period Reg Load from Shadow Select 0: The period register (TBPRD) is loaded from its shadow register when the time-base counter, TBCTR, is equal to zero and/or a sync event as determined by the TBCTL2[PRDLDSYNC] bit. A write/read to the TBPRD register accesses the shadow register. 1: Immediate Mode (Shadow register bypassed): A write or read to the TBPRD register accesses the active register. Reset type: SYSRSn
2	PHSEN	R/W	0h	Counter Reg Load from Phase Reg Enable 0: Do not load the time-base counter (TBCTR) from the time-base phase register (TBPHS). 1: Allow Counter to be loaded from the Phase register (TBPHS) and shadow to active load events when an EPWMxSYNCI input signal occurs or a software-forced sync signal, see bit 6. Reset type: SYSRSn
1-0	CTRMODE	R/W	3h	Counter Mode The time-base counter mode is normally configured once and not changed during normal operation. If you change the mode of the counter, the change will take effect at the next TBCLK edge and the current counter value shall increment or decrement from the value before the mode change. These bits set the time-base counter mode of operation as follows: 00: Up-count mode 01: Down-count mode 10: Up-down count mode 11: Freeze counter operation (default on reset) Reset type: SYSRSn

30.20.2.2 TBCTL2 Register (Offset = 2h) [Reset = 0000h]

TBCTL2 is shown in Figure 30-123 and described in Table 30-28.

Return to the Summary Table.

Time Base Control Register 2

Figure 30-123. TBCTL2 Register

15	14	13	12	11	10	9	8
PRDLDSYNC		RESERVED		RESERVED			
R/W-0h		R-0h		R-0-0h			
7	6	5	4	3	2	1	0
OSHTSYNC	OSHTSYNCMODE	RESERVED	RESERVED				
R-0/W1S-0h	R/W-0h	R/W-0h	R-0-0h				

Table 30-28. TBCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	PRDLDSYNC	R/W	0h	Shadow to Active Period Register Load on SYNC event 00: Shadow to Active Load of TBPRD occurs only when TBCTR = 0 (same as legacy). 01: Shadow to Active Load of TBPRD occurs both when TBCTR = 0 and when SYNC occurs. 10: Shadow to Active Load of TBPRD occurs only when a SYNC is received. 11: Reserved Note: This bit selection is valid only if TBCTL[PRDLD]=0. Reset type: SYSRSn
13-12	RESERVED	R	0h	Reserved
11-8	RESERVED	R-0	0h	Reserved
7	OSHTSYNC	R-0/W1S	0h	Oneshot sync bit 0: Writing a '0' has no effect. 1: Allow one sync pulse to propogate. Reset type: SYSRSn
6	OSHTSYNCMODE	R/W	0h	Oneshot sync enable bit 0: Oneshot sync mode disabled 1: Oneshot sync mode enabled Reset type: SYSRSn
5	RESERVED	R/W	0h	Reserved
4-0	RESERVED	R-0	0h	Reserved

30.20.2.3 EPWMSYNCINSEL Register (Offset = 6h) [Reset = 0001h]

EPWMSYNCINSEL is shown in Figure 30-124 and described in Table 30-29.

Return to the Summary Table.

EPWMxSYNCIN Source Select Register

Figure 30-124. EPWMSYNCINSEL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	SEL						
R-0h	R/W-1h						

Table 30-29. EPWMSYNCINSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6-0	SEL	R/W	1h	These bits determine the source of the EPWMxSYNCI signal. 0x00 Disabled Other Values defined in the 'ePWM SYNC Selection' table Reset type: SYSRSn

30.20.2.4 TBCTR Register (Offset = 8h) [Reset = 0000h]

TBCTR is shown in Figure 30-125 and described in Table 30-30.

Return to the Summary Table.

Time Base Counter Register

Figure 30-125. TBCTR Register

15	14	13	12	11	10	9	8
TBCTR							
R/W-0h							
7	6	5	4	3	2	1	0
TBCTR							
R/W-0h							

Table 30-30. TBCTR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	TBCTR	R/W	0h	Time Base Counter Register Reset type: SYSRSn

30.20.2.5 TBSTS Register (Offset = Ah) [Reset = 0001h]

TBSTS is shown in Figure 30-126 and described in Table 30-31.

Return to the Summary Table.

Time Base Status Register

Figure 30-126. TBSTS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					CTRMAX	SYNCI	CTRDIR
R-0-0h					R/W1C-0h	R/W1C-0h	R-1h

Table 30-31. TBSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-3	RESERVED	R-0	0h	Reserved
2	CTRMAX	R/W1C	0h	Time-Base Counter Max Latched Status Bit 0: Reading a 0 indicates the time-base counter never reached its maximum value. Writing a 0 will have no effect. 1: Reading a 1 on this bit indicates that the time-base counter reached the max value 0xFFFF. Writing a 1 to this bit will clear the latched event. Reset type: SYSRSn
1	SYNCI	R/W1C	0h	Input Synchronization Latched Status Bit 0: Writing a 0 will have no effect. Reading a 0 indicates no external synchronization event has occurred. 1: Reading a 1 on this bit indicates that an external synchronization event has occurred (EPWMxSYNCI). Writing a 1 to this bit will clear the latched event. Reset type: SYSRSn
0	CTRDIR	R	1h	Time Base Counter Direction Status Bit 0: Time-Base Counter is currently counting down. 1: Time-Base Counter is currently counting up. Note: This bit is only valid when the counter is not frozen. Reset type: SYSRSn

30.20.2.6 EPWMSYNCOUTEN Register (Offset = Ch) [Reset = 0001h]

EPWMSYNCOUTEN is shown in Figure 30-127 and described in Table 30-32.

Return to the Summary Table.

EPWMxSYNCOUT Source Enable Register

Figure 30-127. EPWMSYNCOUTEN Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	DCBEVT1EN	DCAEVT1EN	CMPDEN	CMPCEN	CMPBEN	ZEROEN	SWEN
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h

Table 30-32. EPWMSYNCOUTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	Reserved
6	DCBEVT1EN	R/W	0h	This bit enables the DCBEVT1.sync event to set the EPWMxSYNCO signal. 0 Disabled 1 The EPWMxSYNCO signal is pulsed for one PWM clock period upon a DCBEVT1.sync event Reset type: SYSRSn
5	DCAEVT1EN	R/W	0h	This bit enables the DCAEVT1.sync event to set the EPWMxSYNCOUT signal. 0 Disabled 1 The EPWMxSYNCOUT signal is pulsed for one PWM clock period upon a DCAEVT1.sync event Reset type: SYSRSn
4	CMPDEN	R/W	0h	This bit enables the TBCTR = CMPD event to set the EPWMxSYNCO signal. 0 Disabled 1 The EPWMxSYNCO signal is pulsed for one PWM clock period upon a time-base counter equal to counter compare D event (TBCTR = CMPD) Reset type: SYSRSn
3	CMPCEN	R/W	0h	This bit enables the TBCTR = CMPC event to set the EPWMxSYNCO signal. 0 Disabled 1 The EPWMxSYNCO signal is pulsed for one PWM clock period upon a time-base counter equal to counter compare C event (TBCTR = CMPC) Reset type: SYSRSn
2	CMPBEN	R/W	0h	This bit enables the TBCTR = CMPB event to set the EPWMxSYNCO signal. 0 Disabled 1 The EPWMxSYNCO signal is pulsed for one PWM clock period upon a time-base counter equal to counter compare B event (TBCTR = CMPB) Reset type: SYSRSn
1	ZEROEN	R/W	0h	This bit enables the TBCTR = 0x0000 event to set the EPWMxSYNCOUT signal. 0 Disabled 1 The EPWMxSYNCOUT signal is pulsed for one PWM clock period upon the value of TBCTR changing to 0x0000 Reset type: SYSRSn

Table 30-32. EPWMSYNCOUTEN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SWEN	R/W	1h	This bit enables the TBCTL.SWFSYNC bit to set the EPWMxSYNCO signal. 0 Disabled 1 The EPWMxSYNCO signal is pulsed for one PWM clock period when the TBCTL.SWFSYNC bit is set Reset type: SYSRSn

30.20.2.7 TBCTL3 Register (Offset = Eh) [Reset = 0000h]

TBCTL3 is shown in Figure 30-128 and described in Table 30-33.

Return to the Summary Table.

Time Base Control Register 3

Figure 30-128. TBCTL3 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							OSSFRCEN
R-0h							R/W-0h

Table 30-33. TBCTL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R	0h	Reserved
0	OSSFRCEN	R/W	0h	This bit determines which bit sets the EPWMxSYNCOUT One Shot Latch. 0 TBCTL2[OSHTSYNC] sets the One Shot Latch 1 GLDCTL2[OSHTLD] sets the One Shot Latch Reset type: SYSRSn

30.20.2.8 CMPCTL Register (Offset = 10h) [Reset = 0000h]

CMPCTL is shown in Figure 30-129 and described in Table 30-34.

Return to the Summary Table.

Counter Compare Control Register

Figure 30-129. CMPCTL Register

15	14	13	12	11	10	9	8
LINKDUTYHR	RESERVED	LOADBSYNC		LOADASYNC		SHDWBFULL	SHDWAFULL
R/W-0h	R-0-0h	R/W-0h		R/W-0h		R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	SHDWBMODE	RESERVED	SHDWAMODE	LOADBMODE		LOADAMODE	
R-0-0h	R/W-0h	R-0-0h	R/W-0h	R/W-0h		R/W-0h	

Table 30-34. CMPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	LINKDUTYHR	R/W	0h	CMPAHR, CMPBHR Register Linking: 0 PWMA and PWMB outputs generated independently and CMPAHR, CMPBHR are independent values as on Type-4 1 When this bit is set CMPBHR assumes the same value as CMPAHR. This is typically used in complimentary PWM output generation (Section 7 details of the operation) Reset type: SYSRSn
14	RESERVED	R-0	0h	Reserved
13-12	LOADBSYNC	R/W	0h	Shadow to Active CMPB Register Load on SYNC event 00: Shadow to Active Load of CMPB:CMPBHR occurs according to LOADBMODE (bits 1,0) (same as legacy) 01: Shadow to Active Load of CMPB:CMPBHR occurs both according to LOADBMODE bits and when SYNC occurs 10: Shadow to Active Load of CMPB:CMPBHR occurs only when a SYNC is received 11: Reserved Note: This bit is valid only if CMPCTL[SHDWBMODE] = 0. Reset type: SYSRSn
11-10	LOADASYNC	R/W	0h	Shadow to Active CMPA Register Load on SYNC event 00: Shadow to Active Load of CMPA:CMPAHR occurs according to LOADAMODE (bits 1,0) (same as legacy) 01: Shadow to Active Load of CMPA:CMPAHR occurs both according to LOADAMODE bits and when SYNC occurs 10: Shadow to Active Load of CMPA:CMPAHR occurs only when a SYNC is received 11: Reserved Note: This bit is valid only if CMPCTL[SHDWAMODE] = 0. Reset type: SYSRSn
9	SHDWBFULL	R	0h	Counter-compare B (CMPB) Shadow Register Full Status Flag This bit self clears once a loadstrobe occurs. 0: CMPB shadow register not full yet 1: Indicates the CMPB shadow register is full a CPU write will overwrite current shadow value Reset type: SYSRSn

Table 30-34. CMPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	SHDWAFULL	R	0h	Counter-compare A (CMPA) Shadow Register Full Status Flag The flag bit is set when a 32-bit write to CMPA:CMPAHR register or a 16-bit write to CMPA register is made. A 16-bit write to CMPAHR register will not affect the flag. This bit self clears once a load-strobe occurs. 0: CMPA shadow register not full yet 1: Indicates the CMPA shadow register is full, a CPU write will overwrite the current shadow value Reset type: SYSRSn
7	RESERVED	R-0	0h	Reserved
6	SHDWBMODE	R/W	0h	Counter-compare B (CMPB) Register Operating Mode 0: Shadow mode. Operates as a double buffer. All writes via the CPU access the shadow register 1: Immediate mode. Only the active compare B register is used. All writes and reads directly access the active register for immediate compare action Reset type: SYSRSn
5	RESERVED	R-0	0h	Reserved
4	SHDWAMODE	R/W	0h	Counter-compare A (CMPA) Register Operating Mode 0: Shadow mode. Operates as a double buffer. All writes via the CPU access the shadow register 1: Immediate mode. Only the active compare register is used. All writes and reads directly access the active register for immediate compare action Reset type: SYSRSn
3-2	LOADBMODE	R/W	0h	Active Counter-Compare B (CMPB) Load From Shadow Select Mode This bit has no effect in immediate mode (CMPCTL[SHDWBMODE] = 1). 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn
1-0	LOADAMODE	R/W	0h	Active Counter-Compare A (CMPA) Load From Shadow Select Mode This bit has no effect in immediate mode (CMPCTL[SHDWAMODE] = 1). 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn

30.20.2.9 CMPCTL2 Register (Offset = 12h) [Reset = 0000h]

CMPCTL2 is shown in Figure 30-130 and described in Table 30-35.

Return to the Summary Table.

Counter Compare Control Register 2

Figure 30-130. CMPCTL2 Register

15	14	13	12	11	10	9	8
RESERVED		LOADDSYNC		LOADCSYNC		RESERVED	
R-0-0h		R/W-0h		R/W-0h		R-0-0h	
7	6	5	4	3	2	1	0
RESERVED	SHDWDMODE	RESERVED	SHDWCMODE	LOADDMODE		LOADCMODE	
R-0-0h	R/W-0h	R-0-0h	R/W-0h	R/W-0h		R/W-0h	

Table 30-35. CMPCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	RESERVED	R-0	0h	Reserved
13-12	LOADDSYNC	R/W	0h	Shadow to Active CMPD Register Load on SYNC event 00: Shadow to Active Load of CMPD occurs according to LOADDMODE 01: Shadow to Active Load of CMPD occurs both according to LOADDMODE bits and when SYNC occurs 10: Shadow to Active Load of CMPD occurs only when a SYNC is received 11: Reserved Note: This bit is valid only if CMPCTL2[SHDWDMODE] = 0. Reset type: SYSRSn
11-10	LOADCSYNC	R/W	0h	Shadow to Active CMPC Register Load on SYNC event 00: Shadow to Active Load of CMPC occurs according to LOADCMODE 01: Shadow to Active Load of CMPC occurs both according to LOADCMODE bits and when SYNC occurs 10: Shadow to Active Load of CMPC occurs only when a SYNC is received 11: Reserved Note: This bit is valid only if CMPCTL2[SHDWCMODE] = 0. Reset type: SYSRSn
9-7	RESERVED	R-0	0h	Reserved
6	SHDWDMODE	R/W	0h	Counter-Compare D Register Operating Mode 0: Shadow mode - operates as a double buffer. All writes via the CPU access Shadow register. 1: Immediate mode - only the Active compare register is used. All writes/reads via the CPU directly access the Active register for immediate Compare action. Reset type: SYSRSn
5	RESERVED	R-0	0h	Reserved
4	SHDWCMODE	R/W	0h	Counter-Compare C Register Operating Mode 0: Shadow mode - operates as a double buffer. All writes via the CPU access Shadow register. 1: Immediate mode - only the Active compare register is used. All writes/reads via the CPU directly access the Active register for immediate Compare action. Reset type: SYSRSn

Table 30-35. CMPCTL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	LOADDMODE	R/W	0h	Active Counter-Compare D (CMPD) Load from Shadow Select Mode 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: Has no effect in Immediate mode. Reset type: SYSRSn
1-0	LOADCMODE	R/W	0h	Active Counter-Compare C (CMPC) Load from Shadow Select Mode 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: Has no effect in Immediate mode. Reset type: SYSRSn

30.20.2.10 DBCTL Register (Offset = 18h) [Reset = 0000h]

DBCTL is shown in Figure 30-131 and described in Table 30-36.

Return to the Summary Table.

Dead-Band Generator Control Register

Figure 30-131. DBCTL Register

15	14	13	12	11	10	9	8
HALFCYCLE	DEDB_MODE	OUTSWAP		SHDWDBFED MODE	SHDWDBRED MODE	LOADFEDMODE	
R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	
7	6	5	4	3	2	1	0
LOADREDMODE		IN_MODE		POLSEL		OUT_MODE	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 30-36. DBCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	HALFCYCLE	R/W	0h	Half Cycle Clocking Enable Bit 0: Full cycle clocking enabled. The dead-band counters are clocked at the TBCLK rate. 1: Half cycle clocking enabled. The dead-band counters are clocked at TBCLK*2. Reset type: SYSRSn
14	DEDB_MODE	R/W	0h	Dead Band Dual-Edge B Mode Control (S8 switch) 0: Rising edge delay applied to InA/InB as selected by S4 switch (IN-MODE bits) on A signal path only. Falling edge delay applied to InA/InB as selected by S5 switch (INMODE bits) on B signal path only. 1: Rising edge delay and falling edge delay applied to source selected by S4 switch (INMODE bits) and output to B signal path only. Note: When this bit is set to 1, user should always either set OUT_MODE bits such that Apath = InA OR OUTSWAP bits such that OutA=Bpath otherwise, OutA will be invalid. Reset type: SYSRSn
13-12	OUTSWAP	R/W	0h	Dead Band Output Swap Control Bit 13 controls the S6 switch and bit 12 controls the S7 switch. 00: OutA and OutB signals are as defined by OUT-MODE bits. 01: OutA = A-path as defined by OUT-MODE bits. OutB = A-path as defined by OUT-MODE bits (rising edge delay or delay-bypassed A signal path). 10: OutA = B-path as defined by OUT-MODE bits (falling edge delay or delay-bypassed B signal path). OutB = B-path as defined by OUT-MODE bits. 11: OutA = B-path as defined by OUT-MODE bits (falling edge delay or delay-bypassed B signal path). OutB = A-path as defined by OUT-MODE bits (rising edge delay or delay-bypassed A signal path). Reset type: SYSRSn
11	SHDWDBFEDMODE	R/W	0h	FED Dead-Band Load Mode 0: Immediate mode. Only the active DBFED register is used. All writes/reads via the CPU directly access the active register for immediate 'FED dead-band action.' 1: Shadow mode. Operates as a double buffer. All writes via the CPU access Shadow register. Default at Reset is Immediate mode (for compatibility with legacy). Reset type: SYSRSn

Table 30-36. DBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SHDWDBREDMODE	R/W	0h	RED Dead-Band Load Mode 0: Immediate mode. Only the active DBRED register is used. All writes/reads via the CPU directly access the active register for immediate 'RED dead-band action.' 1: Shadow mode. Operates as a double buffer. All writes via the CPU access Shadow register. Default at Reset is Immediate mode (for compatibility with legacy). Reset type: SYSRSn
9-8	LOADFEDMODE	R/W	0h	Active DBFED Load from Shadow Select Mode 00: Load on Counter = 0 (CNT_eq) 01: Load on Counter = Period (PRD_eq) 10: Load on either Counter = 0, or Counter = Period 11: Freeze (no loads possible) Note: has no effect in Immediate mode. Reset type: SYSRSn
7-6	LOADREDMODE	R/W	0h	Active DBRED Load from Shadow Select Mode 00: Load on Counter = 0 (CNT_eq) 01: Load on Counter = Period (PRD_eq) 10: Load on either Counter = 0, or Counter = Period 11: Freeze (no loads possible) Note: has no effect in Immediate mode. Reset type: SYSRSn
5-4	IN_MODE	R/W	0h	Dead-Band Input Mode Control Bit 5 controls the S5 switch and bit 4 controls the S4 switch shown. This allows you to select the input source to the falling-edge and rising-edge delay. To produce classical dead-band waveforms the default is EPWMxA In is the source for both falling and rising-edge delays. 00: EPWMxA In (from the action-qualifier) is the source for both falling-edge and rising-edge delay. 01: EPWMxB In (from the action-qualifier) is the source for rising-edge delayed signal. EPWMxA In (from the action-qualifier) is the source for falling-edge delayed signal. 10: EPWMxA In (from the action-qualifier) is the source for rising-edge delayed signal. EPWMxB In (from the action-qualifier) is the source for falling-edge delayed signal. 11: EPWMxB In (from the action-qualifier) is the source for both rising-edge delay and falling-edge delayed signal. Reset type: SYSRSn
3-2	POLSEL	R/W	0h	Polarity Select Control Bit 3 controls the S3 switch and bit 2 controls the S2 switch. This allows you to selectively invert one of the delayed signals before it is sent out of the dead-band submodule. The following descriptions correspond to classical upper/lower switch control as found in one leg of a digital motor control inverter. These assume that DBCTL[OUT_MODE] = 1,1 and DBCTL[IN_MODE] = 0x0. Other enhanced modes are also possible, but not regarded as typical usage modes. 00: Active high (AH) mode. Neither EPWMxA nor EPWMxB is inverted (default). 01: Active low complementary (ALC) mode. EPWMxA is inverted. 10: Active high complementary (AHC). EPWMxB is inverted. 11: Active low (AL) mode. Both EPWMxA and EPWMxB are inverted. Reset type: SYSRSn

Table 30-36. DBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	OUT_MODE	R/W	0h	Dead-Band Output Mode Control Bit 1 controls the S1 switch and bit 0 controls the S0 switch. 00: DBM is fully disabled or by-passed. In this mode the POLSEL and IN-MODE bits have no effect. 01: Apath = InA (delay is by-passed for A signal path) Bpath = FED (Falling Edge Delay in B signal path) 10: Apath = RED (Rising Edge Delay in A signal path) Bpath = InB (delay is by-passed for B signal path) 11: DBM is fully enabled (i.e. both RED and FED active) Reset type: SYSRSn

30.20.2.11 DBCTL2 Register (Offset = 1Ah) [Reset = 0000h]

DBCTL2 is shown in Figure 30-132 and described in Table 30-37.

Return to the Summary Table.

Dead-Band Generator Control Register 2

Figure 30-132. DBCTL2 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					SHDWDBCTLMODE	LOADDBCTLMODE	
R-0-0h					R/W-0h	R/W-0h	

Table 30-37. DBCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-3	RESERVED	R-0	0h	Reserved
2	SHDWDBCTLMODE	R/W	0h	DBCTL Load Mode 0: Immediate mode - only the Active DBCTL register is used. All writes/reads via the CPU directly access the Active register. 1: Shadow mode - All writes and reads to bits [5:0] of the DBCTL register are shadowed. All other bits still access the active register. Reset type: SYSRSn
1-0	LOADDBCTLMODE	R/W	0h	Active DBCTL Load from Shadow Select Mode 00: Load on Counter = 0 (CNT_eq) 01: Load on Counter = Period (PRD_eq) 10: Load on either Counter = 0, or Counter = Period 11: Freeze (no loads possible) Note: has no effect in Immediate mode Reset type: SYSRSn

30.20.2.12 AQCTL Register (Offset = 20h) [Reset = 0000h]

AQCTL is shown in Figure 30-133 and described in Table 30-38.

Return to the Summary Table.

Action Qualifier Control Register

Figure 30-133. AQCTL Register

15	14	13	12	11	10	9	8
RESERVED				LDAQBSYNC		LDAQASYNC	
R-0-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED	SHDWAQBMODE	RESERVED	SHDWAQAMODE	LDAQBMODE		LDAQAMODE	
R-0-0h	R/W-0h	R-0-0h	R/W-0h	R/W-0h		R/W-0h	

Table 30-38. AQCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R-0	0h	Reserved
11-10	LDAQBSYNC	R/W	0h	Shadow to Active AQCTLB Register Load on SYNC event 00: Shadow to Active Load of AQCTLB occurs according to LDAQBMODE 01: Shadow to Active Load of AQCTLB occurs both according to LDAQBMODE bits and when SYNC occurs. 10: Shadow to Active Load of AQCTLB occurs only when a SYNC is received. 11: Reserved Note: This bit is valid only if AQCTL[SHDWAQBMODE] = 1. Reset type: SYSRSn
9-8	LDAQASYNC	R/W	0h	Shadow to Active AQCTLA Register Load on SYNC event 00: Shadow to Active Load of AQCTLA occurs according to LDAQAMODE 01: Shadow to Active Load of AQCTLA occurs both according to LDAQAMODE bits and when SYNC occurs. 10: Shadow to Active Load of AQCTLA occurs only when a SYNC is received. 11: Reserved Note: This bit is valid only if AQCTL[SHDWAQAMODE] = 1. Reset type: SYSRSn
7	RESERVED	R-0	0h	Reserved
6	SHDWAQBMODE	R/W	0h	Action Qualifier B Register operating mode 1: Shadow mode - operates as a double buffer. All writes via the CPU access Shadow register. 0: Immediate mode - only the Active action qualifier register is used. All writes/reads via the CPU directly access the Active register. Reset type: SYSRSn
5	RESERVED	R-0	0h	Reserved
4	SHDWAQAMODE	R/W	0h	Action Qualifier A Register operating mode 1: Shadow mode - operates as a double buffer. All writes via the CPU access Shadow register. 0: Immediate mode - only the Active action qualifier register is used. All writes/reads via the CPU directly access the Active register. Reset type: SYSRSn

Table 30-38. AQCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	LDAQBMODE	R/W	0h	Active Action Qualifier B Load from Shadow Select Mode 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: has no effect in Immediate mode. Reset type: SYSRSn
1-0	LDAQAMODE	R/W	0h	Active Action Qualifier A Load from Shadow Select Mode 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: has no effect in Immediate mode. Reset type: SYSRSn

30.20.2.13 AQTSRCSEL Register (Offset = 22h) [Reset = 0000h]

AQTSRCSEL is shown in Figure 30-134 and described in Table 30-39.

Return to the Summary Table.

Action Qualifier Trigger Event Source Select Register

Figure 30-134. AQTSRCSEL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
T2SEL				T1SEL			
R/W-0h				R/W-0h			

Table 30-39. AQTSRCSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R-0	0h	Reserved
7-4	T2SEL	R/W	0h	T2 Event Source Select Bits 0000: DCAEVT1 0001: DCAEVT2 0010: DCBEVT1 0011: DCBEVT2 0100: TZ1 0101: TZ2 0110: TZ3 0111: EPWMxSYNCI 1000: DCEVTFILT Others: Reserved Reset type: SYSRSn
3-0	T1SEL	R/W	0h	T1 Event Source Select Bits 0000: DCAEVT1 0001: DCAEVT2 0010: DCBEVT1 0011: DCBEVT2 0100: TZ1 0101: TZ2 0110: TZ3 0111: EPWMxSYNCI 1000: DCEVTFILT Others: Reserved Reset type: SYSRSn

30.20.2.14 PCCTL Register (Offset = 28h) [Reset = 0000h]

PCCTL is shown in Figure 30-135 and described in Table 30-40.

Return to the Summary Table.

PWM Chopper Control Register

Figure 30-135. PCCTL Register

15	14	13	12	11	10	9	8
RESERVED						CHPDUTY	
R-0-0h						R/W-0h	
7	6	5	4	3	2	1	0
CHPFREQ			OSHTWTH				CHPEN
R/W-0h			R/W-0h				R/W-0h

Table 30-40. PCCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R-0	0h	Reserved
10-8	CHPDUTY	R/W	0h	Chopping Clock Duty Cycle 000: Duty = 1/8 (12.5%) 001: Duty = 2/8 (25.0%) 010: Duty = 3/8 (37.5%) 011: Duty = 4/8 (50.0%) 100: Duty = 5/8 (62.5%) 101: Duty = 6/8 (75.0%) 110: Duty = 7/8 (87.5%) 111: Reserved Reset type: SYSRSn
7-5	CHPFREQ	R/W	0h	Chopping Clock Frequency 000: Divide by 1 (no prescale, = 12.5 MHz at 100 MHz TBCLK) 001: Divide by 2 (6.25 MHz at 100 MHz TBCLK) 010: Divide by 3 (4.16 MHz at 100 MHz TBCLK) 011: Divide by 4 (3.12 MHz at 100 MHz TBCLK) 100: Divide by 5 (2.50 MHz at 100 MHz TBCLK) 101: Divide by 6 (2.08 MHz at 100 MHz TBCLK) 110: Divide by 7 (1.78 MHz at 100 MHz TBCLK) 111: Divide by 8 (1.56 MHz at 100 MHz TBCLK) Reset type: SYSRSn
4-1	OSHTWTH	R/W	0h	One-Shot Pulse Width 0000: 1 x EPWMCLK / 8 wide (= 80 ns at 100 MHz EPWMCLK) 0001: 2 x EPWMCLK / 8 wide (= 160 ns at 100 MHz EPWMCLK) 0010: 3 x EPWMCLK / 8 wide (= 240 ns at 100 MHz EPWMCLK) 0011: 4 x EPWMCLK / 8 wide (= 320 ns at 100 MHz EPWMCLK) 0100: 5 x EPWMCLK / 8 wide (= 400 ns at 100 MHz EPWMCLK) 0101: 6 x EPWMCLK / 8 wide (= 480 ns at 100 MHz EPWMCLK) 0110: 7 x EPWMCLK / 8 wide (= 560 ns at 100 MHz EPWMCLK) 0111: 8 x EPWMCLK / 8 wide (= 640 ns at 100 MHz EPWMCLK) 1000: 9 x EPWMCLK / 8 wide (= 720 ns at 100 MHz EPWMCLK) 1001: 10 x EPWMCLK / 8 wide (= 800 ns at 100 MHz EPWMCLK) 1010: 11 x EPWMCLK / 8 wide (= 880 ns at 100 MHz EPWMCLK) 1011: 12 x EPWMCLK / 8 wide (= 960 ns at 100 MHz EPWMCLK) 1100: 13 x EPWMCLK / 8 wide (= 1040 ns at 100 MHz EPWMCLK) 1101: 14 x EPWMCLK / 8 wide (= 1120 ns at 100 MHz EPWMCLK) 1110: 15 x EPWMCLK / 8 wide (= 1200 ns at 100 MHz EPWMCLK) 1111: 16 x EPWMCLK / 8 wide (= 1280 ns at 100 MHz EPWMCLK) Reset type: SYSRSn

Table 30-40. PCCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	CHPEN	R/W	0h	PWM-Chopping Enable 0: Disable (bypass) PWM chopping function 1: Enable chopping function Reset type: SYSRSn

30.20.2.15 VCAPCTL Register (Offset = 30h) [Reset = 0000h]

VCAPCTL is shown in Figure 30-136 and described in Table 30-41.

Return to the Summary Table.

Valley Capture Control Register

Figure 30-136. VCAPCTL Register

15	14	13	12	11	10	9	8
RESERVED					EDGEFILTDLYSEL	VDELAYDIV	
R-0-0h					R/W-0h	R/W-0h	
7	6	5	4	3	2	1	0
VDELAYDIV	RESERVED		TRIGSEL			VCAPSTART	VCAPE
R/W-0h	R-0-0h		R/W-0h			R-0/W1S-0h	R/W-0h

Table 30-41. VCAPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R-0	0h	Reserved
10	EDGEFILTDLYSEL	R/W	0h	Valley Switching Mode Delay Selection 0: No delay applied to the edge filter output 1: HWDELAYVAL delay applied to the edge filter output Reset type: SYSRSn
9-7	VDELAYDIV	R/W	0h	Valley Delay Mode Divide Enable 000: HWVDELVAL = SWVDELVAL 001: HWVDELVAL = VCNTVAL+SWVDELVAL 010: HWVDELVAL = VCNTVAL>>1+SWVDELVAL 011: HWVDELVAL = VCNTVAL>>2+SWVDELVAL 100: HWVDELVAL = VCNTVAL>>4+SWVDELVAL Note: Delay value between the consecutive edge captures can optionally be divided by using these bits. Reset type: SYSRSn
6-5	RESERVED	R-0	0h	Reserved
4-2	TRIGSEL	R/W	0h	Status of Numbered of Captured Events 000: Capture sequence is triggered by software via writes to VCAPCTL[VCAPSTART]. 001: Capture sequence is triggered by CNT_zero event. 010: Capture sequence is triggered by PRD_eq event. 011: Capture sequence is triggered by CNT_zero or PRD_eq event. 100: Capture sequence is triggered by DCAEVT1 event. 101: Capture sequence is triggered by DCAEVT2 event. 110: Capture sequence is triggered by DCBEVT1 event. 111: Capture sequence is triggered by DCBEVT2 event. Note: Valley capture sequence triggered by the selected event in this register field. Once the chosen event occurs the capture sequence is armed. Event captures occur based of the event chosen in DCFCTL[SRCSEL] register. Note: Same event may not be chosen in both DCFCTL[SRCSEL] and VCAPCTL[TRIGSEL] registers. Note: Once the chosen event in VCAPCTL[TRIGSEL] occurs, irrespective of the current capture status, capture sequence is retriggered. Reset type: SYSRSn

Table 30-41. VCAPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	VCAPSTART	R-0/W1S	0h	Valley Capture Start 0: Writing a 0 has no effect 1: Trigger the capture sequence once if VCAPCTL[TRIGSEL]=0x0 Note: This bit is used to start valley capture sequence through software. VCAPCTL[TRIGSEL] has to be chosen for software trigger for this bit to have any effect. Writing of 1 will result in one capture sequence trigger. Reset type: SYSRSn
0	VCAPE	R/W	0h	Valley Capture Enable/Disable 0: Disabled 1: Enabled Reset type: SYSRSn

30.20.2.16 VCNTCFG Register (Offset = 32h) [Reset = 0000h]

VCNTCFG is shown in Figure 30-137 and described in Table 30-42.

Return to the Summary Table.

Valley Counter Config Register

Figure 30-137. VCNTCFG Register

15	14	13	12	11	10	9	8
STOPEDGESTS	RESERVED			STOPEDGE			
R-0h	R-0-0h			R/W-0h			
7	6	5	4	3	2	1	0
STARTEDGESTS	RESERVED			STARTEDGE			
R-0h	R-0-0h			R/W-0h			

Table 30-42. VCNTCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
15	STOPEDGESTS	R	0h	Stop Edge Status Bit 0: Stop edge has not occurred 1: Stop edge occurred Note: This bit is set only after the trigger sequence is armed (upon occurrence of trigger pulse selected through VCAPCTL[TRIGSEL]) and STOPEDGE occurs. Note:This bit is reset by the occurrence of the trigger pulse selected through VCAPCTL[TRIGSEL] Reset type: SYSRSn
14-12	RESERVED	R-0	0h	Reserved
11-8	STOPEDGE	R/W	0h	Counter Stop Edge Selection Once the counter operation is armed, upon occurrence of trigger pulse selected through VCAPCTL[TRIGSEL] pulse - valley counter would stop counting upon the occurrence of chosen number of events thorough this bit field. Stop counting on occurrence of: 0000: Do not stop 0001 1st edge 0010: 2nd edge 0011: 3rd edge ... 1,1,1,1: 15th edge Reset type: SYSRSn
7	STARTEDGESTS	R	0h	Start Edge Status Bit 0: Start edge has not occurred 1: Start edge occurred Note: This bit is set only after the trigger sequence is armed (upon occurrence of trigger pulse selected through VCAPCTL[TRIGSEL]) and STARTEDGE occurs. Note:This bit is reset by the occurrence of the trigger pulse selected through VCAPCTL[TRIGSEL] Reset type: SYSRSn
6-4	RESERVED	R-0	0h	Reserved

Table 30-42. VCNTCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	STARTEDGE	R/W	0h	Counter Start Edge Selection Once the counter operation is armed, upon occurrence of trigger pulse selected through VCAPCTL[TRIGSEL] pulse - valley counter would start counting upon the occurrence of chosen number of events thorough this bit field. Start counting on occurrence of 0000: Do not start 0001: 1st edge 0010: 2nd edge 0011: 3rd edge ... 1111: 15th edge Reset type: SYSRSn

30.20.2.17 HRCNFG Register (Offset = 40h) [Reset = 0000h]

HRCNFG is shown in Figure 30-138 and described in Table 30-43.

Return to the Summary Table.

HRPWM Configuration Register
This register is only accessible on EPWM modules with HRPWM capabilities.

Figure 30-138. HRCNFG Register

15	14	13	12	11	10	9	8
RESERVED		RESERVED	HRLOADB		CTLMODEB	EDGMODEB	
R/W-0h		R-0-0h	R/W-0h		R/W-0h	R/W-0h	
7	6	5	4	3	2	1	0
SWAPAB	AUTOCONV	SELOUTB	HRLOAD		CTLMODE	EDGMODE	
R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	

Table 30-43. HRCNFG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	RESERVED	R/W	0h	Reserved
13	RESERVED	R-0	0h	Reserved
12-11	HRLOADB	R/W	0h	Shadow Mode Bit Selects the time event that loads the CMPBHR shadow value into the active register. 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Load on CMPB_EQ (Translator Event CMPB-3) Reset type: SYSRSn
10	CTLMODEB	R/W	0h	Control Mode Bits Selects the register (CMP/TBPRD or TBPHS) that controls the MEP: 0: CMPBHR(8) or TBPRDHR(8) Register controls the edge position (i.e., this is duty or period control mode). (Default on Reset) 1: TBPHSHR(8) Register controls the edge position (i.e., this is phase control mode). Reset type: SYSRSn
9-8	EDGMODEB	R/W	0h	Edge Mode Bits Selects the edge of the PWM that is controlled by the micro-edge position (MEP) logic: 00: HRPWM capability is disabled (default on reset) 01: MEP control of rising edge (CMPBHR) 10: MEP control of falling edge (CMPBHR) 11: MEP control of both edges (TBPHSHR or TBPRDHR) Reset type: SYSRSn
7	SWAPAB	R/W	0h	Swap ePWM A & B Output Signals This bit enables the swapping of the A & B signal outputs. The selection is as follows: 0: ePWMxA and ePWMxB outputs are unchanged. 1: ePWMxA signal appears on ePWMxB output and ePWMxB signal appears on ePWMxA output. Reset type: SYSRSn

Table 30-43. HRCNFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	AUTOCONV	R/W	0h	Auto Convert Delay Line Value Selects whether the fractional duty cycle/period/phase in the CMPAHR/TBPRDHR/TBPHSHR register is automatically scaled by the MEP scale factor in the HRMSTEP register or manually scaled by calculations in application software. The SFO library function automatically updates the HRMSTEP register with the appropriate MEP scale factor. 0: Automatic HRMSTEP scaling is disabled. 1: Automatic HRMSTEP scaling is enabled. If application software is manually scaling the fractional duty cycle, or phase (i.e. software sets CMPAHR = (fraction(PWMduty * PWMperiod) * MEP Scale Factor)<<8 + 0x080 for duty cycle), then this mode must be disabled. Reset type: SYSRSn
5	SELOUTB	R/W	0h	EPWMxB Output Select Bit This bit selects which signal is output on the ePWMxB channel output. The inversion will take the high resolution mode into account and the inverted signal will contain any high resolution modification. The inversion takes place as the last step in modifying the ePWMxB signal. 0: ePWMxB output is normal. 1: ePWMxB output is inverted version of ePWMxA signal. Reset type: SYSRSn
4-3	HRLOAD	R/W	0h	Shadow Mode Bit Selects the time event that loads the CMPAHR shadow value into the active register. 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Load on CMPA_EQ (Translator Event CMPA-3) Reset type: SYSRSn
2	CTLMODE	R/W	0h	Control Mode Bits Selects the register (CMP/TBPRD or TBPHS) that controls the MEP: 0: CMPAHR(8) or TBPRDHR(8) Register controls the edge position (i.e., this is duty or period control mode). (Default on Reset) 1: TBPHSHR(8) Register controls the edge position (i.e., this is phase control mode). Reset type: SYSRSn
1-0	EDGMODE	R/W	0h	Edge Mode Bits Selects the edge of the PWM that is controlled by the micro-edge position (MEP) logic: 00: HRPWM capability is disabled (default on reset) 01: MEP control of rising edge (CMPAHR) 10: MEP control of falling edge (CMPAHR) 11: MEP control of both edges (TBPHSHR or TBPRDHR) Reset type: SYSRSn

30.20.2.18 HRCNFG2 Register (Offset = 4Eh) [Reset = 0000h]

HRCNFG2 is shown in Figure 30-139 and described in Table 30-44.

Return to the Summary Table.

HRPWM Configuration 2 Register
This register is only accessible on EPWM modules with HRPWM capabilities. Only 16 bit accesses are allowed for this register. Debugger access in 32 bit mode may display incorrect values.

Figure 30-139. HRCNFG2 Register

15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED					
R/W-0h	R-0/W1S-0h	R-0-0h					
7	6	5	4	3	2	1	0
RESERVED		CTLMODEDBFED		CTLMODEDBRED		EDGMODEDB	
R-0-0h		R/W-0h		R/W-0h		R/W-0h	

Table 30-44. HRCNFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R-0/W1S	0h	Reserved
13-6	RESERVED	R-0	0h	Reserved
5-4	CTLMODEDBFED	R/W	0h	Shadow Mode Bit - selection should match DBCTL[LOADFEDMODE] Selects the time event that loads the DBFEDHR shadow value into the active register. 00 Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01 Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10 Load on either CTR = Zero or CTR = PRD 11 Reserved Reset type: SYSRSn
3-2	CTLMODEDBRED	R/W	0h	Shadow Mode Bit - selection should match DBCTL[LOADREDMODE] Selects the time event that loads the DBREDHR shadow value into the active register. 00 Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01 Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10 Load on either CTR = Zero or CTR = PRD 11 Reserved Reset type: SYSRSn
1-0	EDGMODEDB	R/W	0h	Edge Mode Bits Selects the edge of the PWM that is controlled by the micro-edge position (MEP) logic: 00 HRPWM capability is disabled (default on reset) 01 MEP control of rising edge (DBREDHR) 10 MEP control of falling edge (DBFEDHR) 11 MEP control of both edges (rising edge of DBREDHR or falling edge of DBFEDHR) Reset type: SYSRSn

30.20.2.19 HRPCTL Register (Offset = 5Ah) [Reset = 0000h]

HRPCTL is shown in Figure 30-140 and described in Table 30-45.

Return to the Summary Table.

High Resolution Period Control Register
Fields in this register related to HRPWM are only applicable on EPWM modules with HRPWM capabilities.

Figure 30-140. HRPCTL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	PWMSYNCSELX			RESERVED	TBPHSHRLOADE	PWMSYNCSEL	HRPE
R-0-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-45. HRPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R-0	0h	Reserved
6-4	PWMSYNCSELX	R/W	0h	Extended selection bits for EPWMSYNCPER 000: EPWMSYNCPER is defined by PWMSYNCSEL - > default condition (compatible with previous EPWM versions) 001: Reserved 010: Reserved 011: Reserved 100: CTR = CMPC, Count direction Up 101: CTR = CMPC, Count direction Down 110: CTR = CMPD, Count direction Up 111: CTR = CMPD, Count direction Down Reset type: SYSRSn
3	RESERVED	R/W	0h	Reserved
2	TBPHSHRLOADE	R/W	0h	TBPHSHR Load Enable This bit allows you to synchronize ePWM modules with a high-resolution phase on a SYNCIN, TBCTL[SWFSYNC] or digital compare event. This allows for multiple ePWM modules operating at the same frequency to be phase aligned with high-resolution. 0: Disables synchronization of high-resolution phase on a SYNCIN, TBCTL[SWFSYNC] or digital compare event: 1: Synchronize the high-resolution phase on a SYNCIN, TBCTL[SWFSYNC] or digital comparator synchronization event. The phase is synchronized using the contents of the high-resolution phase TBPHSHR register. The TBCTL[PHSEN] bit which enables the loading of the TBCTR register with TBPHS register value on a SYNCIN or TBCTL[SWFSYNC] event works independently. However, users need to enable this bit also if they want to control phase in conjunction with the high-resolution period feature. This bit and the TBCTL[PHSEN] bit must be set to 1 when high-resolution period is enabled for up-down count mode even if TBPHSHR = 0x0000. This bit does not need to be set when only high-resolution duty is enabled. Reset type: SYSRSn
1	PWMSYNCSEL	R/W	0h	PWMSYNC Source Select Bit: This bit selects the source for the EPWMSYNCPER signal that goes to the CMPSS and GPDAC: 0 CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 1 CTR = zero: Time-base counter equal to zero (TBCTR = 0x00) Reset type: SYSRSn

Table 30-45. HRPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	HRPE	R/W	0h	High Resolution Period Enable Bit 0: High resolution period feature disabled. In this mode the ePWM behaves as a Type 4 ePWM. 1: High resolution period enabled. In this mode the HRPWM module can control high-resolution of both the duty and frequency. When high-resolution period is enabled, TBCTL[CTRMODE] = 0,1 (down-count mode) is not supported. Reset type: SYSRSn

30.20.2.20 TRREM Register (Offset = 5Ch) [Reset = 0000h]

TRREM is shown in Figure 30-141 and described in Table 30-46.

Return to the Summary Table.

HRPWM High Resolution Remainder Register
This register is only accessible on EPWM modules with HRPWM capabilities.

Figure 30-141. TRREM Register

15	14	13	12	11	10	9	8
RESERVED					TRREM		
R-0-0h					R/W-0h		
7	6	5	4	3	2	1	0
TRREM							
R/W-0h							

Table 30-46. TRREM Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R-0	0h	Reserved
10-0	TRREM	R/W	0h	HRPWM Remainder Bits: This 11-bit value keeps track of the remainder portion of the HRPWM algorithm calculations. This value keeps track of the remainder portion of the HRPWM hardware calculations. Notes: 1. The lower 8-bits of the TRREM register can be automatically initialized with the TBPHSHR value on a SYNCIN or TBCTL[SWFSYNC] event or DC event (if enabled). The user can also write a value with the CPU. 2. Priority of TRREM register updates: Sync (software or hardware) TBPHSHR copied to TRREM : Highest Priority HRPWM Hardware (updates TRREM register): Next priority CPU Write To TRREM Register: Lowest Priority 3. Bit 10 of TRREM register is not used in asymmetrical mode. This bit can be forced to zero. TRREM will be initialized to 0x0 and 0x100 in Up and Up-down modes respectively. Asymmetrical Mode: TRREM[7:0] = TBPHSHR[15:8] TRREM[10,9,8] = 0,0,0 Symmetrical Mode: TRREM[7:0] = TBPHSHR[15:8] TRREM[10,9,8] = 0,0,1 Reset type: SYSRSn

30.20.2.21 GLDCTL Register (Offset = 68h) [Reset = 0000h]

GLDCTL is shown in Figure 30-142 and described in Table 30-47.

Return to the Summary Table.

Global PWM Load Control Register

Figure 30-142. GLDCTL Register

15	14	13	12	11	10	9	8
RESERVED			GLDCNT			GLDPRD	
R-0-0h			R-0h			R/W-0h	
7	6	5	4	3	2	1	0
GLDPRD	RESERVED	OSHTMODE	GLDMODE				GLD
R/W-0h	R-0-0h	R/W-0h	R/W-0h				R/W-0h

Table 30-47. GLDCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R-0	0h	Reserved
12-10	GLDCNT	R	0h	Global Load Strobe Counter Register These bits indicate how many selected events have occurred: 000: No events 001: 1 event 010: 2 events 011: 3 events 100: 4 events 101: 5 events 110: 6 events 111: 7 events Reset type: SYSRSn
9-7	GLDPRD	R/W	0h	Global Load Strobe Period Select Register These bits select how many selected events need to occur before a load strobe is generated 000: Disable counter 001: Generate strobe on GLDCNT = 001 (1st event) 010: Generate strobe on GLDCNT = 010 (2nd event) 011: Generate strobe on GLDCNT = 011 (3rd event) 100: Generate strobe on GLDCNT = 100 (4th event) 101: Generate strobe on GLDCNT = 101 (5th event) 110: Generate strobe on GLDCNT = 110 (6th event) 111: Generate strobe on GLDCNT = 111 (7th event) Reset type: SYSRSn
6	RESERVED	R-0	0h	Reserved
5	OSHTMODE	R/W	0h	One Shot Load Mode Control Bit 0: One shot load mode is disabled and shadow to active loading happens continuously on all the chosen load strobes. 1: One shot mode is active. All load strobes are blocked until GLDCTL2[OSHTLD] is written with 1. Note: One Shot mode can only be used with global shadow to active load mode enabled (GLDCTL[GLD]=1) Reset type: SYSRSn

Table 30-47. GLDCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-1	GLDMODE	R/W	0h	Global Load Pulse selection for Shadow to Active Mode Reloads 0000: Load on Counter = 0 (CNT_ZRO) 0001: Load on Counter = Period (PRD_EQ) 0010: Load on either Counter = 0, or Counter = Period 0011: Load on SYNCEVT - this is logical OR of DCAEVT1.sync, DCBEVT1.sync, EPWMxSYNCI and TBCTL[SWFSYNC] 0100: Load on SYNCEVT or CNT_ZRO 0101: Load on SYNCEVT or PRD_EQ 0110: Load on SYNCEVT or CNT_ZRO or PRD_EQ 1000: Load on Counter = CMPCU (CMPC_EQ counter incrementing) 1001: Load on Counter = CMPCD (CMPC_EQ counter decrementing) 1010: Load on Counter = CMPDU (CMPD_EQ counter incrementing) 1011: Load on Counter = CMPDD (CMPD_EQ counter decrementing) 1100: Reserved ... 1110: Reserved 1111: Load on GLDCTL2[GFRCLD] write Reset type: SYSRSn
0	GLD	R/W	0h	Global Shadow to Active Load Event Control 0: Shadow to active reload for all shadowed registers happens as per the individual reload control bits specified (Compatible with previous EPWM versions). 1: When set, all the shadow to active reload events are defined by GLDMODE bits in GLDCTL register. All the shadow registers use same reload pulse from shadow to active reloading. Individual LOADMODE bits are ignored. Reset type: SYSRSn

30.20.2.22 GLDCFG Register (Offset = 6Ah) [Reset = 0000h]

GLDCFG is shown in Figure 30-143 and described in Table 30-48.

Return to the Summary Table.

Global PWM Load Config Register

Figure 30-143. GLDCFG Register

15	14	13	12	11	10	9	8
RESERVED					AQCSFRC	AQCTLB_AQCTLB2	AQCTLA_AQCTLA2
R-0-0h					R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DBCTL	DBFED_DBFEDHR	DBRED_DBREDHR	CMPD	CMPC	CMPB_CMPBHR	CMPA_CMPAHR	TBPRD_TBPRDHR
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-48. GLDCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R-0	0h	Reserved
10	AQCSFRC	R/W	0h	Global load event configuration for AQCSFRC 0: Registers use local reload configuration even if GLDCTL(GLD)=1 (reload is compatible with previous EPWMs) 1: Registers use global load configuration if this bit is set and GLDCTL(GLD)=1 Reset type: SYSRSn
9	AQCTLB_AQCTLB2	R/W	0h	Global load event configuration for AQCTLB_AQCTLB2 0: Registers use local reload configuration even if GLDCTL(GLD)=1 (reload is compatible with previous EPWMs) 1: Registers use global load configuration if this bit is set and GLDCTL(GLD)=1 Reset type: SYSRSn
8	AQCTLA_AQCTLA2	R/W	0h	Global load event configuration for AQCTLA_AQCTLA2 0: Registers use local reload configuration even if GLDCTL(GLD)=1 (reload is compatible with previous EPWMs) 1: Registers use global load configuration if this bit is set and GLDCTL(GLD)=1 Reset type: SYSRSn
7	DBCTL	R/W	0h	Global load event configuration for DBCTL 0: Registers use local reload configuration even if GLDCTL(GLD)=1 (reload is compatible with previous EPWMs) 1: Registers use global load configuration if this bit is set and GLDCTL(GLD)=1 Reset type: SYSRSn
6	DBFED_DBFEDHR	R/W	0h	Global load event configuration for DBFED_DBFEDHR 0: Registers use local reload configuration even if GLDCTL(GLD)=1 (reload is compatible with previous EPWMs) 1: Registers use global load configuration if this bit is set and GLDCTL(GLD)=1 Reset type: SYSRSn
5	DBRED_DBREDHR	R/W	0h	Global load event configuration for DBRED_DBREDHR 0: Registers use local reload configuration even if GLDCTL(GLD)=1 (reload is compatible with previous EPWMs) 1: Registers use global load configuration if this bit is set and GLDCTL(GLD)=1 Reset type: SYSRSn

Table 30-48. GLDCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CMPD	R/W	0h	Global load event configuration for CMPD 0: Registers use local reload configuration even if GLDCTL(GLD)=1 (reload is compatible with previous EPWMs) 1: Registers use global load configuration if this bit is set and GLDCTL(GLD)=1 Reset type: SYSRSn
3	CMPC	R/W	0h	Global load event configuration for CMPC 0: Registers use local reload configuration even if GLDCTL(GLD)=1 (reload is compatible with previous EPWMs) 1: Registers use global load configuration if this bit is set and GLDCTL(GLD)=1 Reset type: SYSRSn
2	CMPB_CMPBHR	R/W	0h	Global load event configuration for CMPB_CMPBHR 0: Registers use local reload configuration even if GLDCTL(GLD)=1 (reload is compatible with previous EPWMs) 1: Registers use global load configuration if this bit is set and GLDCTL(GLD)=1 Reset type: SYSRSn
1	CMPA_CMPAHR	R/W	0h	Global load event configuration for CMPA_CMPAHR 0: Registers use local reload configuration even if GLDCTL(GLD)=1 (reload is compatible with previous EPWMs) 1: Registers use global load configuration if this bit is set and GLDCTL(GLD)=1 Reset type: SYSRSn
0	TBPRD_TBPRDHR	R/W	0h	Global load event configuration for TBPRD_TBPRDHR 0: Registers use local reload configuration even if GLDCTL(GLD)=1 (reload is compatible with previous EPWMs) 1: Registers use global load configuration if this bit is set and GLDCTL(GLD)=1 Reset type: SYSRSn

30.20.2.23 AQCTLA Register (Offset = 80h) [Reset = 0000h]

AQCTLA is shown in Figure 30-144 and described in Table 30-49.

Return to the Summary Table.

Action Qualifier Control Register For Output A

Figure 30-144. AQCTLA Register

15	14	13	12	11	10	9	8
RESERVED				CBD		CBU	
R-0-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
CAD		CAU		PRD		ZRO	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 30-49. AQCTLA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxA output low. 10: Set: force EPWMxA output high. 11: Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxA output low. 10: Set: force EPWMxA output high. 11: Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxA output low. 10: Set: force EPWMxA output high. 11: Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxA output low. 10: Set: force EPWMxA output high. 11: Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 30-49. AQCTLA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxA output low. 10: Set: force EPWMxA output high. 11: Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxA output low. 10: Set: force EPWMxA output high. 11: Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

30.20.2.24 AQCTLA2 Register (Offset = 82h) [Reset = 0000h]

AQCTLA2 is shown in Figure 30-145 and described in Table 30-50.

Return to the Summary Table.

Additional Action Qualifier Control Register For Output A

Figure 30-145. AQCTLA2 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
T2D		T2U		T1D		T1U	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 30-50. AQCTLA2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R-0	0h	Reserved
7-6	T2D	R/W	0h	Action when event occurs on T2 in DOWN-Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxA output low. 10: Set: force EPWMxA output high. 11: Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	T2U	R/W	0h	Action when event occurs on T2 in UP-Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxA output low. 10: Set: force EPWMxA output high. 11: Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
3-2	T1D	R/W	0h	Action when event occurs on T1 in DOWN-Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxA output low. 10: Set: force EPWMxA output high. 11: Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	T1U	R/W	0h	Action when event occurs on T1 in UP-Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxA output low. 10: Set: force EPWMxA output high. 11: Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

30.20.2.25 AQCTLB Register (Offset = 84h) [Reset = 0000h]

AQCTLB is shown in Figure 30-146 and described in Table 30-51.

Return to the Summary Table.

Action Qualifier Control Register For Output B

Figure 30-146. AQCTLB Register

15	14	13	12	11	10	9	8
RESERVED				CBD		CBU	
R-0-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
CAD		CAU		PRD		ZRO	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 30-51. AQCTLB Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxB output low. 10: Set: force EPWMxB output high. 11: Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxB output low. 10: Set: force EPWMxB output high. 11: Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxB output low. 10: Set: force EPWMxB output high. 11: Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxB output low. 10: Set: force EPWMxB output high. 11: Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 30-51. AQCTLB Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxB output low. 10: Set: force EPWMxB output high. 11: Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxB output low. 10: Set: force EPWMxB output high. 11: Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

30.20.2.26 AQCTLB2 Register (Offset = 86h) [Reset = 0000h]

AQCTLB2 is shown in Figure 30-147 and described in Table 30-52.

Return to the Summary Table.

Additional Action Qualifier Control Register For Output B

Figure 30-147. AQCTLB2 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
T2D		T2U		T1D		T1U	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 30-52. AQCTLB2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R-0	0h	Reserved
7-6	T2D	R/W	0h	Action when event occurs on T2 in DOWN-Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxB output low. 10: Set: force EPWMxB output high. 11: Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	T2U	R/W	0h	Action when event occurs on T2 in UP-Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxB output low. 10: Set: force EPWMxB output high. 11: Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
3-2	T1D	R/W	0h	Action when event occurs on T1 in DOWN-Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxB output low. 10: Set: force EPWMxB output high. 11: Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	T1U	R/W	0h	Action when event occurs on T1 in UP-Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force EPWMxB output low. 10: Set: force EPWMxB output high. 11: Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

30.20.2.27 AQSFRC Register (Offset = 8Eh) [Reset = 0000h]

AQSFRC is shown in Figure 30-148 and described in Table 30-53.

Return to the Summary Table.

Action Qualifier Software Force Register

Figure 30-148. AQSFRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RLDCSF		OTSFB	ACTSFB		OTSFA	ACTSFA	
R/W-0h		R-0/W1S-0h	R/W-0h		R-0/W1S-0h	R/W-0h	

Table 30-53. AQSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R-0	0h	Reserved
7-6	RLDCSF	R/W	0h	AQCSFRC Active Register Reload From Shadow Options 00: Load on time-base counter equals zero 01: Load on time-base counter equals period 10: Load on time-base counter equals zero or counter equals period 11: Load immediately (the active register is directly accessed by the CPU and is not loaded from the shadow register). Reset type: SYSRSn
5	OTSFB	R-0/W1S	0h	One-Time Software Forced Event on Output B 0: Writing a 0 (zero) has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (i.e., a forced event is initiated.). This is a one-shot forced event. It can be overridden by another subsequent event on output B. 1: Initiates a single software forced event Reset type: SYSRSn
4-3	ACTSFB	R/W	0h	Action When One-Time Software Force B is Invoked 00: Does nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Note: This action is not qualified by counter direction (CNT_dir) Reset type: SYSRSn
2	OTSFA	R-0/W1S	0h	One-Time Software Forced Event on Output A 0: Writing a 0 (zero) has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (i.e., a forced event is initiated). This is a one-shot forced event. It can be overridden by another subsequent event on output A. 1: Initiates a single software forced event Reset type: SYSRSn
1-0	ACTSFA	R/W	0h	Action When One-Time Software Force A Is Invoked 00: Does nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Note: This action is not qualified by counter direction (CNT_dir) Reset type: SYSRSn

30.20.2.28 AQCSFRC Register (Offset = 92h) [Reset = 0000h]

AQCSFRC is shown in Figure 30-149 and described in Table 30-54.

Return to the Summary Table.

Action Qualifier Continuous S/W Force Register

Figure 30-149. AQCSFRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				CSFB		CSFA	
R-0-0h				R/W-0h		R/W-0h	

Table 30-54. AQCSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3-2	CSFB	R/W	0h	Continuous Software Force on Output B In immediate mode, a continuous force takes effect on the next TBCLK edge. In shadow mode, a continuous force takes effect on the next TBCLK edge after a shadow load into the active register. To configure shadow mode, use AQSFRC[RLDCSF]. 00: Software forcing is disabled and has no effect 01: Forces a continuous low on output B 10: Forces a continuous high on output B 11: Software forcing is disabled and has no effect Reset type: SYSRSn
1-0	CSFA	R/W	0h	Continuous Software Force on Output A In immediate mode, a continuous force takes effect on the next TBCLK edge. In shadow mode, a continuous force takes effect on the next TBCLK edge after a shadow load into the active register. 00: Software forcing is disabled and has no effect 01: Forces a continuous low on output A 10: Forces a continuous high on output A 11: Software forcing is disabled and has no effect Reset type: SYSRSn

30.20.2.29 DBREDHR Register (Offset = A0h) [Reset = 0000h]

DBREDHR is shown in Figure 30-150 and described in Table 30-55.

Return to the Summary Table.

Dead-Band Generator Rising Edge Delay High Resolution Mirror Register

Figure 30-150. DBREDHR Register

15	14	13	12	11	10	9	8
DBREDHR							RESERVED
R/W-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED							RESERVED
R-0h							R-0h

Table 30-55. DBREDHR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	DBREDHR	R/W	0h	Dead Band Rising Edge Delay High Resolution Bits Reset type: SYSRSn
8	RESERVED	R	0h	Reserved
7-1	RESERVED	R	0h	Reserved
0	RESERVED	R	0h	Reserved

30.20.2.30 DBRED Register (Offset = A2h) [Reset = 0000h]

DBRED is shown in Figure 30-151 and described in Table 30-56.

Return to the Summary Table.

Dead-Band Generator Rising Edge Delay High Resolution Mirror Register

Figure 30-151. DBRED Register

15	14	13	12	11	10	9	8
RESERVED		DBRED					
R-0h		R/W-0h					
7	6	5	4	3	2	1	0
DBRED							
R/W-0h							

Table 30-56. DBRED Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	RESERVED	R	0h	Reserved
13-0	DBRED	R/W	0h	Rising edge delay value Reset type: SYSRSn

30.20.2.31 DBFEDHR Register (Offset = A4h) [Reset = 0000h]

DBFEDHR is shown in Figure 30-152 and described in Table 30-57.

Return to the Summary Table.

Dead-Band Generator Falling Edge Delay High Resolution Register

Figure 30-152. DBFEDHR Register

15	14	13	12	11	10	9	8
DBFEDHR							RESERVED
R/W-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED							RESERVED
R-0h							R-0h

Table 30-57. DBFEDHR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	DBFEDHR	R/W	0h	Dead Band Falling Edge Delay High Resolution Bits Reset type: SYSRSn
8	RESERVED	R	0h	Reserved
7-1	RESERVED	R	0h	Reserved
0	RESERVED	R	0h	Reserved

30.20.2.32 DBFED Register (Offset = A6h) [Reset = 0000h]

DBFED is shown in Figure 30-153 and described in Table 30-58.

Return to the Summary Table.

Dead-Band Generator Falling Edge Delay Count Register

Figure 30-153. DBFED Register

15	14	13	12	11	10	9	8
RESERVED			DBFED				
R-0h			R/W-0h				
7	6	5	4	3	2	1	0
DBFED							
R/W-0h							

Table 30-58. DBFED Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	RESERVED	R	0h	Reserved
13-0	DBFED	R/W	0h	Falling Edge Delay Count 14-bit counter Reset type: SYSRSn

30.20.2.33 TBPHS Register (Offset = C0h) [Reset = 00000000h]

TBPHS is shown in Figure 30-154 and described in Table 30-59.

Return to the Summary Table.

Time Base Phase High

Figure 30-154. TBPHS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPHS																TBPHSHR															
R/W-0h																R/W-0h															

Table 30-59. TBPHS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TBPHS	R/W	0h	Phase Offset Register These bits set time-base counter phase of the selected ePWM relative to the time-base that is supplying the synchronization input signal. - If TBCTL[PHSEN] = 0, then the synchronization event is ignored and the time-base counter is not loaded with the phase. - If TBCTL[PHSEN] = 1, then the time-base counter (TBCTR) will be loaded with the phase (TBPHS) when a synchronization event occurs. The synchronization event can be initiated by the input synchronization signal (EPWMxSYNCI) or by a software forced synchronization. Reset type: SYSRSn
15-0	TBPHSHR	R/W	0h	Phase Offset (High Resolution) Register. TBPHSHR must not be used. Instead TRREM (HRPWM remainder register) must be used to mimic the functionality of TBPHSHR. The lower 8 bits in this register are ignored - writes are ignored and reads return zero Reset type: SYSRSn

30.20.2.34 TBPRDHR Register (Offset = C4h) [Reset = 0000h]

TBPRDHR is shown in Figure 30-155 and described in Table 30-60.

Return to the Summary Table.

Time Base Period High Resolution Register

Figure 30-155. TBPRDHR Register

15	14	13	12	11	10	9	8
TBPRDHR							
R/W-0h							
7	6	5	4	3	2	1	0
TBPRDHR							
R/W-0h							

Table 30-60. TBPRDHR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	TBPRDHR	R/W	0h	Period High Resolution Bits The upper 8-bits contain the high-resolution portion of the period value. The TBPRDHR register is not affected by the TBCTL[PRDLD] bit. Reads from this register always reflect the shadow register. Likewise writes are also to the shadow register. The TBPRDHR register is only used when the high resolution period feature is enabled. This register is only available with ePWM modules which support high-resolution period control. The lower 8 bits in this register are ignored - writes are ignored and reads return zero Reset type: SYSRSn

30.20.2.35 TBPRD Register (Offset = C6h) [Reset = 0000h]

TBPRD is shown in Figure 30-156 and described in Table 30-61.

Return to the Summary Table.

Time Base Period Register

Figure 30-156. TBPRD Register

15	14	13	12	11	10	9	8
TBPRD							
R/W-0h							
7	6	5	4	3	2	1	0
TBPRD							
R/W-0h							

Table 30-61. TBPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	TBPRD	R/W	0h	Time Base Period Register These bits determine the period of the time-base counter. This sets the PWM frequency. Shadowing of this register is enabled and disabled by the TBCTL[PRDLD] bit. By default this register is shadowed. - If TBCTL[PRDLD] = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the active register will be loaded from the shadow register when the time-base counter equals zero. - If TBCTL[PRDLD] = 1, then the shadow is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. - The active and shadow registers share the same memory map address. Reset type: SYSRSn

30.20.2.36 CMPA Register (Offset = D4h) [Reset = 00000000h]

CMPA is shown in Figure 30-157 and described in Table 30-62.

Return to the Summary Table.

Counter Compare A Register

Figure 30-157. CMPA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPA																CMPAHR															
R/W-0h																R/W-0h															

Table 30-62. CMPA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CMPA	R/W	0h	Compare A Register The value in the active CMPA register is continuously compared to the time-base counter (TBCTR). When the values are equal, the counter-compare module generates a 'time-base counter equal to counter compare A' event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the EPWMxA or the EPWMxB output depending on the configuration of the AQCTLA and AQCTLB registers. The actions that can be defined in the AQCTLA and AQCTLB registers include: - Do nothing the event is ignored. - Clear: Pull the EPWMxA and/or EPWMxB signal low - Set: Pull the EPWMxA and/or EPWMxB signal high - Toggle the EPWMxA and/or EPWMxB signal Shadowing of this register is enabled and disabled by the CMPCTL[SHDWAMODE] bit. By default this register is shadowed. - If CMPCTL[SHDWAMODE] = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the CMPCTL[LOADAMODE] bit field determines which event will load the active register from the shadow register. - Before a write, the CMPCTL[SHDWAFULL] bit can be read to determine if the shadow register is currently full. - If CMPCTL[SHDWAMODE] = 1, then the shadow register is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. - In either mode, the active and shadow registers share the same memory map address. Reset type: SYSRSn
15-0	CMPAHR	R/W	0h	Compare A HRPWM Extension Register The UPPER 8-bits contain the high-resolution portion (most significant 8-bits) of the counter-compare A value. CMPA:CMPAHR can be accessed in a single 32-bit read/write. Shadowing is enabled and disabled by the CMPCTL[SHDWAMODE] bit as described for the CMPA register. The lower 8 bits in this register are ignored Reset type: SYSRSn

30.20.2.37 CMPB Register (Offset = D8h) [Reset = 00000000h]

CMPB is shown in Figure 30-158 and described in Table 30-63.

Return to the Summary Table.

Compare B Register

Figure 30-158. CMPB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPB																CMPBHR															
R/W-0h																R/W-0h															

Table 30-63. CMPB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CMPB	R/W	0h	Compare B Register The value in the active CMPB register is continuously compared to the time-base counter (TBCTR). When the values are equal, the counter-compare module generates a 'time-base counter equal to counter compare B' event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the EPWMxA or the EPWMxB output depending on the configuration of the AQCTLA and AQCTLB registers. The actions that can be defined in the AQCTLA and AQCTLB registers include: - Do nothing the event is ignored. - Clear: Pull the EPWMxA and/or EPWMxB signal low - Set: Pull the EPWMxA and/or EPWMxB signal high - Toggle the EPWMxA and/or EPWMxB signal Shadowing of this register is enabled and disabled by the CMPCTL[SHDWBMODE] bit. By default this register is shadowed. - If CMPCTL[SHDWBMODE] = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the CMPCTL[LOADBMODE] bit field determines which event will load the active register from the shadow register. - Before a write, the CMPCTL[SHDWBFULL] bit can be read to determine if the shadow register is currently full. - If CMPCTL[SHDWBMODE] = 1, then the shadow register is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. - In either mode, the active and shadow registers share the same memory map address. Reset type: SYSRSn
15-0	CMPBHR	R/W	0h	Compare B High Resolution Bits The lower 8 bits in this register are ignored Reset type: SYSRSn

30.20.2.38 CMPC Register (Offset = DEh) [Reset = 0000h]

CMPC is shown in Figure 30-159 and described in Table 30-64.

Return to the Summary Table.

Counter Compare C Register
LINK feature access should always be 16-bit

Figure 30-159. CMPC Register

15	14	13	12	11	10	9	8
CMPC							
R/W-0h							
7	6	5	4	3	2	1	0
CMPC							
R/W-0h							

Table 30-64. CMPC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CMPC	R/W	0h	Compare C Register The value in the active CMPC register is continuously compared to the time-base counter (TBCTR). When the values are equal, the counter-compare module generates a 'time-base counter equal to counter compare C' event. Shadowing of this register is enabled and disabled by the CMPCTL2[SHDWCMODE] bit. By default this register is shadowed. - If CMPCTL2[SHDWCMODE] = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the CMPCTL2[LOADCMODE] bit field determines which event will load the active register from the shadow register: - If CMPCTL2[SHDWCMODE] = 1, then the shadow register is disabled and any write or read will go directly to the active register that is, the register actively controlling the hardware. - In either mode, the active and shadow registers share the same memory map address. Reset type: SYSRSn

30.20.2.39 CMPD Register (Offset = E2h) [Reset = 0000h]

CMPD is shown in Figure 30-160 and described in Table 30-65.

Return to the Summary Table.

Counter Compare D Register
LINK feature access should always be 16-bit

Figure 30-160. CMPD Register

15	14	13	12	11	10	9	8
CMPD							
R/W-0h							
7	6	5	4	3	2	1	0
CMPD							
R/W-0h							

Table 30-65. CMPD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CMPD	R/W	0h	Compare D Register The value in the active CMPD register is continuously compared to the time-base counter (TBCTR). When the values are equal, the counter-compare module generates a 'time-base counter equal to counter compare D' event. Shadowing of this register is enabled and disabled by the CMPCTL2[SHDWDMODE] bit. By default this register is shadowed. - If CMPCTL2[SHDWDMODE] = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the CMPCTL2[LOADDMODE] bit field determines which event will load the active register from the shadow register: - If CMPCTL2[SHDWDMODE] = 1, then the shadow register is disabled and any write or read will go directly to the active register that is, the register actively controlling the hardware. - In either mode, the active and shadow registers share the same memory map address. Reset type: SYSRSn

30.20.2.40 GLDCTL2 Register (Offset = E8h) [Reset = 0000h]

GLDCTL2 is shown in Figure 30-161 and described in Table 30-66.

Return to the Summary Table.

Global PWM Load Control Register 2

Figure 30-161. GLDCTL2 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						GFRCLD	OSHTLD
R-0-0h						R-0/W1S-0h	R-0/W1S-0h

Table 30-66. GLDCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-2	RESERVED	R-0	0h	Reserved
1	GFRCLD	R-0/W1S	0h	Force Load Event in One Shot Mode 0: Writing of 0 will be ignored. Always reads back a 0. 1: Force one load event at the input of the event pre-scale counter. This bit is intended to be used for testing and/or software force loading of the events in global load mode. Reset type: SYSRSn
0	OSHTLD	R-0/W1S	0h	Enable Reload Event in One Shot Mode 0: Writing of 0 will be ignored. Always reads back a 0. 1: Turns the one shot latch condition ON. Upon occurrence of a chosen load strobe, one shadow to active reload occurs and the latch will be cleared. Hence writing 1 to this bit would allow one load strobe event to pass through and block further strobe events. Reset type: SYSRSn

30.20.2.41 SWVDELVAL Register (Offset = EEh) [Reset = 0000h]

SWVDELVAL is shown in Figure 30-162 and described in Table 30-67.

Return to the Summary Table.

Software Valley Mode Delay Register

Figure 30-162. SWVDELVAL Register

15	14	13	12	11	10	9	8
SWVDELVAL							
R/W-0h							
7	6	5	4	3	2	1	0
SWVDELVAL							
R/W-0h							

Table 30-67. SWVDELVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	SWVDELVAL	R/W	0h	Software Valley Delay Value Register This register can be optionally used define offset value for the hardware calculated delay HWDELAYVAL as defined in VCAPCTL[VDELAYDIV] bits. Reset type: SYSRSn

30.20.2.42 TZSEL Register (Offset = 100h) [Reset = 0000h]

TZSEL is shown in Figure 30-163 and described in Table 30-68.

Return to the Summary Table.

Trip Zone Select Register

Figure 30-163. TZSEL Register

15	14	13	12	11	10	9	8
DCBEVT1	DCAEVT1	OSHT6	OSHT5	OSHT4	OSHT3	OSHT2	OSHT1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DCBEVT2	DCAEVT2	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-68. TZSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	DCBEVT1	R/W	0h	Digital Compare Output B Event 1 Select 0: Disable DCBEVT1 as one-shot-trip source for this ePWM module. 1: Enable DCBEVT1 as one-shot-trip source for this ePWM module. Reset type: SYSRSn
14	DCAEVT1	R/W	0h	Digital Compare Output A Event 1 Select 0: Disable DCAEVT1 as one-shot-trip source for this ePWM module. 1: Enable DCAEVT1 as one-shot-trip source for this ePWM module. Reset type: SYSRSn
13	OSHT6	R/W	0h	Trip-zone 6 (TZ6) Select 0: Disable TZ6 as a one-shot trip source for this ePWM module 1: Enable TZ6 as a one-shot trip source for this ePWM module Reset type: SYSRSn
12	OSHT5	R/W	0h	Trip-zone 5 (TZ5) Select 0: Disable TZ5 as a one-shot trip source for this ePWM module 1: Enable TZ5 as a one-shot trip source for this ePWM module Reset type: SYSRSn
11	OSHT4	R/W	0h	Trip-zone 4 (TZ4) Select 0: Disable TZ4 as a one-shot trip source for this ePWM module 1: Enable TZ4 as a one-shot trip source for this ePWM module Reset type: SYSRSn
10	OSHT3	R/W	0h	Trip-zone 3 (TZ3) Select 0: Disable TZ3 as a one-shot trip source for this ePWM module 1: Enable TZ3 as a one-shot trip source for this ePWM module Reset type: SYSRSn
9	OSHT2	R/W	0h	Trip-zone 2 (TZ2) Select 0: Disable TZ2 as a one-shot trip source for this ePWM module 1: Enable TZ2 as a one-shot trip source for this ePWM module Reset type: SYSRSn
8	OSHT1	R/W	0h	Trip-zone 1 (TZ1) Select 0: Disable TZ1 as a one-shot trip source for this ePWM module 1: Enable TZ1 as a one-shot trip source for this ePWM module Reset type: SYSRSn
7	DCBEVT2	R/W	0h	Digital Compare Output B Event 2 Select 0: Disable DCBEVT2 as a CBC trip source for this ePWM module 1: Enable DCBEVT2 as a CBC trip source for this ePWM module Reset type: SYSRSn

Table 30-68. TZSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	DCAEVT2	R/W	0h	Digital Compare Output A Event 2 Select 0: Disable DCAEVT2 as a CBC trip source for this ePWM module 1: Enable DCAEVT2 as a CBC trip source for this ePWM module Reset type: SYSRSn
5	CBC6	R/W	0h	Trip-zone 6 (TZ6) Select 0: Disable TZ6 as a CBC trip source for this ePWM module 1: Enable TZ6 as a CBC trip source for this ePWM module Reset type: SYSRSn
4	CBC5	R/W	0h	Trip-zone 5 (TZ5) Select 0: Disable TZ5 as a CBC trip source for this ePWM module 1: Enable TZ5 as a CBC trip source for this ePWM module Reset type: SYSRSn
3	CBC4	R/W	0h	Trip-zone 4 (TZ4) Select 0: Disable TZ4 as a CBC trip source for this ePWM module 1: Enable TZ4 as a CBC trip source for this ePWM module Reset type: SYSRSn
2	CBC3	R/W	0h	Trip-zone 3 (TZ3) Select 0: Disable TZ3 as a CBC trip source for this ePWM module 1: Enable TZ3 as a CBC trip source for this ePWM module Reset type: SYSRSn
1	CBC2	R/W	0h	Trip-zone 2 (TZ2) Select 0: Disable TZ2 as a CBC trip source for this ePWM module 1: Enable TZ2 as a CBC trip source for this ePWM module Reset type: SYSRSn
0	CBC1	R/W	0h	Trip-zone 1 (TZ1) Select 0: Disable TZ1 as a CBC trip source for this ePWM module 1: Enable TZ1 as a CBC trip source for this ePWM module Reset type: SYSRSn

30.20.2.43 TZSEL2 Register (Offset = 102h) [Reset = 0000h]

TZSEL2 is shown in Figure 30-164 and described in Table 30-69.

Return to the Summary Table.

Trip Zone Select Register 2

Figure 30-164. TZSEL2 Register

15	14	13	12	11	10	9	8
RESERVED							CAPEVTOST
R-0-0h							R/W-0h
7	6	5	4	3	2	1	0
RESERVED							CAPEVTCBC
R-0-0h							R/W-0h

Table 30-69. TZSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8	CAPEVTOST	R/W	0h	CAPEVT OST Select 0: Disable CAPEVT as a one-shot trip source for this ePWM module 1: Enable CAPEVT as a one-shot trip source for this ePWM module Reset type: SYSRSn
7-1	RESERVED	R-0	0h	Reserved
0	CAPEVTCBC	R/W	0h	CAPEVT CBC mode Select 0: Disable CAPEVT as a CBC trip source for this ePWM module 1: Enable CAPEVT as a CBC trip source for this ePWM module Reset type: SYSRSn

30.20.2.44 TZDCSEL Register (Offset = 104h) [Reset = 0000h]

TZDCSEL is shown in Figure 30-165 and described in Table 30-70.

Return to the Summary Table.

Trip Zone Digital Comparator Select Register

Figure 30-165. TZDCSEL Register

15	14	13	12	11	10	9	8
RESERVED				DCBEVT2			DCBEVT1
R-0-0h				R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
DCBEVT1		DCAEVT2			DCAEVT1		
R/W-0h		R/W-0h			R/W-0h		

Table 30-70. TZDCSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R-0	0h	Reserved
11-9	DCBEVT2	R/W	0h	Digital Compare Output B Event 2 Selection 000: Event disabled 001: DCBH = low, DCBL = don't care 010: DCBH = high, DCBL = don't care 011: DCBL = low, DCBH = don't care 100: DCBL = high, DCBH = don't care 101: DCBL = high, DCBH = low 110: Reserved 111: Reserved Reset type: SYSRSn
8-6	DCBEVT1	R/W	0h	Digital Compare Output B Event 1 Selection 000: Event disabled 001: DCBH = low, DCBL = don't care 010: DCBH = high, DCBL = don't care 011: DCBL = low, DCBH = don't care 100: DCBL = high, DCBH = don't care 101: DCBL = high, DCBH = low 110: Reserved 111: Reserved Reset type: SYSRSn
5-3	DCAEVT2	R/W	0h	Digital Compare Output A Event 2 Selection 000: Event disabled 001: DCAH = low, DCAL = don't care 010: DCAH = high, DCAL = don't care 011: DCAL = low, DCAH = don't care 100: DCAL = high, DCAH = don't care 101: DCAL = high, DCAH = low 110: Reserved 111: Reserved Reset type: SYSRSn
2-0	DCAEVT1	R/W	0h	Digital Compare Output A Event 1 Selection 000: Event disabled 001: DCAH = low, DCAL = don't care 010: DCAH = high, DCAL = don't care 011: DCAL = low, DCAH = don't care 100: DCAL = high, DCAH = don't care 101: DCAL = high, DCAH = low 110: Reserved 111: Reserved Reset type: SYSRSn

30.20.2.45 TZCTL Register (Offset = 108h) [Reset = 0000h]

TZCTL is shown in Figure 30-166 and described in Table 30-71.

Return to the Summary Table.

Trip Zone Control Register

Figure 30-166. TZCTL Register

15	14	13	12	11	10	9	8
RESERVED				DCBEVT2		DCBEVT1	
R-0-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
DCAEVT2		DCAEVT1		TZB		TZA	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 30-71. TZCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R-0	0h	Reserved
11-10	DCBEVT2	R/W	0h	Digital Compare Output B Event 2 Action On EPWMxB 00: High-impedance (EPWMxB = High-impedance state) 01: Force EPWMxB to a high state. 10: Force EPWMxB to a low state. 11: Do Nothing, trip action is disabled Reset type: SYSRSn
9-8	DCBEVT1	R/W	0h	Digital Compare Output B Event 1 Action On EPWMxB 00: High-impedance (EPWMxB = High-impedance state) 01: Force EPWMxB to a high state. 10: Force EPWMxB to a low state. 11: Do Nothing, trip action is disabled Reset type: SYSRSn
7-6	DCAEVT2	R/W	0h	Digital Compare Output A Event 2 Action On EPWMxA 00: High-impedance (EPWMxA = High-impedance state) 01: Force EPWMxA to a high state. 10: Force EPWMxA to a low state. 11: Do Nothing, trip action is disabled Reset type: SYSRSn
5-4	DCAEVT1	R/W	0h	Digital Compare Output A Event 1 Action On EPWMxA 00: High-impedance (EPWMxA = High-impedance state) 01: Force EPWMxA to a high state. 10: Force EPWMxA to a low state. 11: Do Nothing, trip action is disabled Reset type: SYSRSn
3-2	TZB	R/W	0h	TZ1 to TZ6, DCAEVT1/2, DCBEVT1/2Trip Action On EPWMxB When a trip event occurs the following action is taken on output EPWMxB. Which trip-zone pins can cause an event is defined in the TZSEL register. 00: High-impedance (EPWMxB = High-impedance state) 01: Force EPWMxB to a high state 10: Force EPWMxB to a low state 11: Do nothing, no action is taken on EPWMxB. Reset type: SYSRSn

Table 30-71. TZCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	TZA	R/W	0h	TZ1 to TZ6, DCAEVT1/2, DCBEVT1/2 Trip Action On EPWMxA When a trip event occurs the following action is taken on output EPWMxA. Which trip-zone pins can cause an event is defined in the TZSEL register. 00: High-impedance (EPWMxA = High-impedance state) 01: Force EPWMxA to a high state 10: Force EPWMxA to a low state 11: Do nothing, no action is taken on EPWMxA. Reset type: SYSRSn

30.20.2.46 TZCTL2 Register (Offset = 10Ah) [Reset = 0000h]

TZCTL2 is shown in Figure 30-167 and described in Table 30-72.

Return to the Summary Table.

Additional Trip Zone Control Register

Figure 30-167. TZCTL2 Register

15	14	13	12	11	10	9	8
ETZE	RESERVED			TZBD			TZBU
R/W-0h	R-0-0h			R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
TZBU		TZAD			TZAU		
R/W-0h		R/W-0h			R/W-0h		

Table 30-72. TZCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	ETZE	R/W	0h	TZCTL2 Enable 0: Use trip action from TZCTL (legacy EPWM compatibility) 1: Use trip action defined in TZCTL2, TZCTLDCA and TZCTLDCB. Settings in TZCTL are ignored Reset type: SYSRSn
14-12	RESERVED	R-0	0h	Reserved
11-9	TZBD	R/W	0h	TZ1 to TZ6 Trip Action On EPWMxB while Count direction is DOWN 000: HiZ (EPWMxB = HiZ state) 001: Forced Hi (EPWMxB = High state) 010: Forced Lo (EPWMxB = Lo state) 011: Toggle (Low -> High, High -> Low) 100: Reserved 101: Reserved 110: Reserved 111: Do Nothing, trip action is disabled Reset type: SYSRSn
8-6	TZBU	R/W	0h	TZ1 to TZ6, DCAEVT1/2, DCBEVT1/2 Trip Action On EPWMxB while Count direction is UP 000: HiZ (EPWMxB = HiZ state) 001: Forced Hi (EPWMxB = High state) 010: Forced Lo (EPWMxB = Lo state) 011: Toggle (Low -> High, High -> Low) 100: Reserved 101: Reserved 110: Reserved 111: Do Nothing, trip action is disabled Reset type: SYSRSn
5-3	TZAD	R/W	0h	TZ1 to TZ6, DCAEVT1/2, DCBEVT1/2 Trip Action On EPWMxA while Count direction is DOWN 000: HiZ (EPWMxA = HiZ state) 001: Forced Hi (EPWMxA = High state) 010: Forced Lo (EPWMxA = Lo state) 011: Toggle (Low -> High, High -> Low) 100: Reserved 101: Reserved 110: Reserved 111: Do Nothing, trip action is disabled Reset type: SYSRSn

Table 30-72. TZCTL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	TZAU	R/W	0h	TZ1 to TZ6, DCAEVT1/2, DCBEVT1/2 Trip Action On EPWMxA while Count direction is UP 000: HiZ (EPWMxA = HiZ state) 001: Forced Hi (EPWMxA = High state) 010: Forced Lo (EPWMxA = Lo state) 011: Toggle (Low -> High, High -> Low) 100: Reserved 101: Reserved 110: Reserved 111: Do Nothing, trip action is disabled Reset type: SYSRSn

30.20.2.47 TZCTLDCA Register (Offset = 10Ch) [Reset = 0000h]

TZCTLDCA is shown in Figure 30-168 and described in Table 30-73.

Return to the Summary Table.

Trip Zone Control Register Digital Compare A

Figure 30-168. TZCTLDCA Register

15	14	13	12	11	10	9	8
RESERVED				DCAEVT2D			DCAEVT2U
R-0-0h				R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
DCAEVT2U		DCAEVT1D			DCAEVT1U		
R/W-0h		R/W-0h			R/W-0h		

Table 30-73. TZCTLDCA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R-0	0h	Reserved
11-9	DCAEVT2D	R/W	0h	Digital Compare Output A Event 2 Action On EPWMxA while Count direction is DOWN 000: HiZ (EPWMxA = HiZ state) 001: Forced Hi (EPWMxA = High state) 010: Forced Lo (EPWMxA = Lo state) 011: Toggle (Low -> High, High -> Low) 100: Reserved 101: Reserved 110: Reserved 111: Do Nothing, trip action is disabled Reset type: SYSRSn
8-6	DCAEVT2U	R/W	0h	Digital Compare Output A Event 2 Action On EPWMxA while Count direction is UP 000: HiZ (EPWMxA = HiZ state) 001: Forced Hi (EPWMxA = High state) 010: Forced Lo (EPWMxA = Lo state) 011: Toggle (Low -> High, High -> Low) 100: Reserved 101: Reserved 110: Reserved 111: Do Nothing, trip action is disabled Reset type: SYSRSn
5-3	DCAEVT1D	R/W	0h	Digital Compare Output A Event 1 Action On EPWMxA while Count direction is DOWN 000: HiZ (EPWMxA = HiZ state) 001: Forced Hi (EPWMxA = High state) 010: Forced Lo (EPWMxA = Lo state) 011: Toggle (Low -> High, High -> Low) 100: Reserved 101: Reserved 110: Reserved 111: Do Nothing, trip action is disabled Reset type: SYSRSn

Table 30-73. TZCTLDCA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	DCAEVT1U	R/W	0h	Digital Compare Output A Event 1 Action On EPWMxA while Count direction is UP 000: HiZ (EPWMxA = HiZ state) 001: Forced Hi (EPWMxA = High state) 010: Forced Lo (EPWMxA = Lo state) 011: Toggle (Low -> High, High -> Low) 100: Reserved 101: Reserved 110: Reserved 111: Do Nothing, trip action is disabled Reset type: SYSRSn

30.20.2.48 TZCTLDCB Register (Offset = 10Eh) [Reset = 0000h]

TZCTLDCB is shown in Figure 30-169 and described in Table 30-74.

Return to the Summary Table.

Trip Zone Control Register Digital Compare B

Figure 30-169. TZCTLDCB Register

15	14	13	12	11	10	9	8
RESERVED				DCBEVT2D			DCBEVT2U
R-0-0h				R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
DCBEVT2U		DCBEVT1D			DCBEVT1U		
R/W-0h		R/W-0h			R/W-0h		

Table 30-74. TZCTLDCB Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R-0	0h	Reserved
11-9	DCBEVT2D	R/W	0h	Digital Compare Output B Event 2 Action On EPWMxB while Count direction is DOWN 000: HiZ (EPWMxB = HiZ state) 001: Forced Hi (EPWMxB = High state) 010: Forced Lo (EPWMxB = Lo state) 011: Toggle (Low -> High, High -> Low) 100: Reserved 101: Reserved 110: Reserved 111: Do Nothing, trip action is disabled Reset type: SYSRSn
8-6	DCBEVT2U	R/W	0h	Digital Compare Output B Event 2 Action On EPWMxB while Count direction is UP 000: HiZ (EPWMxB = HiZ state) 001: Forced Hi (EPWMxB = High state) 010: Forced Lo (EPWMxB = Lo state) 011: Toggle (Low -> High, High -> Low) 100: Reserved 101: Reserved 110: Reserved 111: Do Nothing, trip action is disabled Reset type: SYSRSn
5-3	DCBEVT1D	R/W	0h	Digital Compare Output B Event 1 Action On EPWMxB while Count direction is DOWN 000: HiZ (EPWMxB = HiZ state) 001: Forced Hi (EPWMxB = High state) 010: Forced Lo (EPWMxB = Lo state) 011: Toggle (Low -> High, High -> Low) 100: Reserved 101: Reserved 110: Reserved 111: Do Nothing, trip action is disabled Reset type: SYSRSn

Table 30-74. TZCTLDCB Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	DCBEVT1U	R/W	0h	Digital Compare Output B Event 1 Action On EPWMxB while Count direction is UP 000: HiZ (EPWMxB = HiZ state) 001: Forced Hi (EPWMxB = High state) 010: Forced Lo (EPWMxB = Lo state) 011: Toggle (Low -> High, High -> Low) 100: Reserved 101: Reserved 110: Reserved 111: Do Nothing, trip action is disabled Reset type: SYSRSn

30.20.2.49 TZEINT Register (Offset = 11Ah) [Reset = 0000h]

TZEINT is shown in Figure 30-170 and described in Table 30-75.

Return to the Summary Table.

Trip Zone Enable Interrupt Register

Figure 30-170. TZEINT Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
CAPEVT	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0-0h

Table 30-75. TZEINT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R-0	0h	Reserved
7	CAPEVT	R/W	0h	Capture Event Interrupt Enable 0: Disabled 1: Enabled Reset type: SYSRSn
6	DCBEVT2	R/W	0h	Digital Compare Output B Event 2 Interrupt Enable 0: Disabled 1: Enabled Reset type: SYSRSn
5	DCBEVT1	R/W	0h	Digital Compare Output B Event 1 Interrupt Enable 0: Disabled 1: Enabled Reset type: SYSRSn
4	DCAEVT2	R/W	0h	Digital Compare Output A Event 2 Interrupt Enable 0: Disabled 1: Enabled Reset type: SYSRSn
3	DCAEVT1	R/W	0h	Digital Compare Output A Event 1 Interrupt Enable 0: Disabled 1: Enabled Reset type: SYSRSn
2	OST	R/W	0h	Trip-zone One-Shot Interrupt Enable 0: Disable one-shot interrupt generation 1: Enable Interrupt generation a one-shot trip event will cause a EPWMx_TZINT PIE interrupt. Reset type: SYSRSn
1	CBC	R/W	0h	Trip-zone Cycle-by-Cycle Interrupt Enable 0: Disable cycle-by-cycle interrupt generation. 1: Enable interrupt generation a cycle-by-cycle trip event will cause an EPWMx_TZINT PIE interrupt. Reset type: SYSRSn
0	RESERVED	R-0	0h	Reserved

30.20.2.50 TZFLG Register (Offset = 126h) [Reset = 0000h]

TZFLG is shown in Figure 30-171 and described in Table 30-76.

Return to the Summary Table.

Trip Zone Flag Register

Figure 30-171. TZFLG Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
CAPEVT	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	INT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 30-76. TZFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R-0	0h	Reserved
7	CAPEVT	R	0h	Latched Status Flag for Capture Event 0: Indicates no trip event has occurred on CAPEVT 1: Indicates a trip event has occurred for the event defined for CAPEVT Reset type: SYSRSn
6	DCBEVT2	R	0h	Latched Status Flag for Digital Compare Output B Event 2 0: Indicates no trip event has occurred on DCBEVT2 1: Indicates a trip event has occurred for the event defined for DCBEVT2 Reset type: SYSRSn
5	DCBEVT1	R	0h	Latched Status Flag for Digital Compare Output B Event 1 0: Indicates no trip event has occurred on DCBEVT1 1: Indicates a trip event has occurred for the event defined for DCBEVT1 Reset type: SYSRSn
4	DCAEVT2	R	0h	Latched Status Flag for Digital Compare Output A Event 2 0: Indicates no trip event has occurred on DCAEVT2 1: Indicates a trip event has occurred for the event defined for DCAEVT2 Reset type: SYSRSn
3	DCAEVT1	R	0h	Latched Status Flag for Digital Compare Output A Event 1 0: Indicates no trip event has occurred on DCAEVT1 1: Indicates a trip event has occurred for the event defined for DCAEVT1 Reset type: SYSRSn
2	OST	R	0h	Latched Status Flag for A One-Shot Trip Event 0: No one-shot trip event has occurred. 1: Indicates a trip event has occurred on a pin selected as a one-shot trip source. This bit is cleared by writing the appropriate value to the TZCLR register. Reset type: SYSRSn

Table 30-76. TZFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	CBC	R	0h	Latched Status Flag for Cycle-By-Cycle Trip Event 0: No cycle-by-cycle trip event has occurred. 1: Indicates a trip event has occurred on a signal selected as a cycle-by-cycle trip source. The TZFLG[CBC] bit will remain set until it is manually cleared by the user. If the cycle-by-cycle trip event is still present when the CBC bit is cleared, then CBC will be immediately set again. The specified condition on the signal is automatically cleared when the ePWM time-base counter reaches zero (TBCTR = 0x00) if the trip condition is no longer present. The condition on the signal is only cleared when the TBCTR = 0x00 no matter where in the cycle the CBC flag is cleared. This bit is cleared by writing the appropriate value to the TZCLR register. Reset type: SYSRSn
0	INT	R	0h	Latched Trip Interrupt Status Flag 0: Indicates no interrupt has been generated. 1: Indicates an EPWMx_TZINT PIE interrupt was generated because of a trip condition. No further EPWMx_TZINT PIE interrupts will be generated until this flag is cleared. If the interrupt flag is cleared when either CBC or OST is set, then another interrupt pulse will be generated. Clearing all flag bits will prevent further interrupts. This bit is cleared by writing the appropriate value to the TZCLR register. Reset type: SYSRSn

30.20.2.51 TZCBCFLG Register (Offset = 128h) [Reset = 0000h]

TZCBCFLG is shown in Figure 30-172 and described in Table 30-77.

Return to the Summary Table.

Trip Zone CBC Flag Register

Figure 30-172. TZCBCFLG Register

15	14	13	12	11	10	9	8
RESERVED							CAPEVT
R-0-0h							R-0h
7	6	5	4	3	2	1	0
DCBEVT2	DCAEVT2	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 30-77. TZCBCFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8	CAPEVT	R	0h	Latched Status Flag for Capture Event 0: Indicates no trip event has occurred on CAPEVT 1: Indicates a trip event has occurred for the event defined for CAPEVT Reset type: SYSRSn
7	DCBEVT2	R	0h	Latched Status Flag for Digital Compare B Output Event 2 Trip Latch 0: Reading a 0 indicates that no trip has occurred on DCBEVT2. 1: Reading a 1 indicates a trip has occured on the DCBEVT2 selected event. Reset type: SYSRSn
6	DCAEVT2	R	0h	Latched Status Flag for Digital Compare A Output Event 2 Trip Latch 0: Reading a 0 indicates that no trip has occurred on DCAEVT2. 1: Reading a 1 indicates a trip has occured on the DCAEVT2 selected event. Reset type: SYSRSn
5	CBC6	R	0h	Latched Status Flag for CBC6 Trip Latch 0: Reading a 0 indicates that no trip has occurred on CBC6. 1: Reading a 1 indicates a trip has occured on the CBC6 selected event. Reset type: SYSRSn
4	CBC5	R	0h	Latched Status Flag for CBC5 Trip Latch 0: Reading a 0 indicates that no trip has occurred on CBC5. 1: Reading a 1 indicates a trip has occured on the CBC5 selected event. Reset type: SYSRSn
3	CBC4	R	0h	Latched Status Flag for CBC4 Trip Latch 0: Reading a 0 indicates that no trip has occurred on CBC4. 1: Reading a 1 indicates a trip has occured on the CBC4 selected event. Reset type: SYSRSn
2	CBC3	R	0h	Latched Status Flag for CBC3 Trip Latch 0: Reading a 0 indicates that no trip has occurred on CBC3. 1: Reading a 1 indicates a trip has occured on the CBC3 selected event. Reset type: SYSRSn

Table 30-77. TZCBCFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	CBC2	R	0h	Latched Status Flag for CBC2 Trip Latch 0: Reading a 0 indicates that no trip has occurred on CBC2. 1: Reading a 1 indicates a trip has occured on the CBC2 selected event. Reset type: SYSRSn
0	CBC1	R	0h	Latched Status Flag for CBC1 Trip Latch 0: Reading a 0 indicates that no trip has occurred on CBC1. 1: Reading a 1 indicates a trip has occured on the CBC1 selected event. Reset type: SYSRSn

30.20.2.52 TZOSTFLG Register (Offset = 12Ah) [Reset = 0000h]

TZOSTFLG is shown in Figure 30-173 and described in Table 30-78.

Return to the Summary Table.

Trip Zone OST Flag Register

Figure 30-173. TZOSTFLG Register

15	14	13	12	11	10	9	8
RESERVED							CAPEVT
R-0-0h							R-0h
7	6	5	4	3	2	1	0
DCBEVT1	DCAEVT1	OST6	OST5	OST4	OST3	OST2	OST1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 30-78. TZOSTFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8	CAPEVT	R	0h	Latched Status Flag for Capture Event 0: Indicates no trip event has occurred on CAPEVT 1: Indicates a trip event has occurred for the event defined for CAPEVT Reset type: SYSRSn
7	DCBEVT1	R	0h	Latched Status Flag for Digital Compare B Output Event 1 Trip Latch 0: Reading a 0 indicates that no trip has occurred on DCBEVT1. 1: Reading a 1 indicates a trip has occured on the DCBEVT1 selected event. Reset type: SYSRSn
6	DCAEVT1	R	0h	Latched Status Flag for Digital Compare A Output Event 1 Trip Latch 0: Reading a 0 indicates that no trip has occurred on DCAEVT1. 1: Reading a 1 indicates a trip has occured on the DCAEVT1 selected event. Reset type: SYSRSn
5	OST6	R	0h	Latched Status Flag for OST6 Trip Latch 0: Reading a 0 indicates that no trip has occurred on OST6. 1: Reading a 1 indicates a trip has occured on the OST6 selected event. Reset type: SYSRSn
4	OST5	R	0h	Latched Status Flag for OST5 Trip Latch 0: Reading a 0 indicates that no trip has occurred on OST5. 1: Reading a 1 indicates a trip has occured on the OST5 selected event. Reset type: SYSRSn
3	OST4	R	0h	Latched Status Flag for OST4 Trip Latch 0: Reading a 0 indicates that no trip has occurred on OST4. 1: Reading a 1 indicates a trip has occured on the OST4 selected event. Reset type: SYSRSn
2	OST3	R	0h	Latched Status Flag for OST3 Trip Latch 0: Reading a 0 indicates that no trip has occurred on OST3. 1: Reading a 1 indicates a trip has occured on the OST3 selected event. Reset type: SYSRSn

Table 30-78. TZOSTFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	OST2	R	0h	Latched Status Flag for OST2 Trip Latch 0: Reading a 0 indicates that no trip has occurred on OST2. 1: Reading a 1 indicates a trip has occured on the OST2 selected event. Reset type: SYSRSn
0	OST1	R	0h	Latched Status Flag for OST1 Trip Latch 0: Reading a 0 indicates that no trip has occurred on OST1. 1: Reading a 1 indicates a trip has occured on the OST1 selected event. Reset type: SYSRSn

30.20.2.53 TZCLR Register (Offset = 12Eh) [Reset = 0000h]

TZCLR is shown in Figure 30-174 and described in Table 30-79.

Return to the Summary Table.

Trip Zone Clear Register

Figure 30-174. TZCLR Register

15	14	13	12	11	10	9	8
CBCPULSE		RESERVED					
R/W-0h		R-0-0h					
7	6	5	4	3	2	1	0
CAPEVT	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	INT
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 30-79. TZCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	CBCPULSE	R/W	0h	Clear Pulse for Cycle-By-Cycle (CBC) Trip Latch This bit field determines which pulse clears the CBC trip latch. 00: CTR = zero pulse clears CBC trip latch. (Same as legacy designs.) 01: CTR = PRD pulse clears CBC trip latch. 10: CTR = zero or CTR = PRD pulse clears CBC trip latch. 11: CBC trip latch is not cleared Reset type: SYSRSn
13-8	RESERVED	R-0	0h	Reserved
7	CAPEVT	R-0/W1S	0h	Clear Flag for Capture Event 0: Writing 0 has no effect. This bit always reads back 0. 1: Writing 1 clears the CAPEVT event trip condition. Reset type: SYSRSn
6	DCBEVT2	R-0/W1S	0h	Clear Flag for Digital Compare Output B Event 2 0: Writing 0 has no effect. This bit always reads back 0. 1: Writing 1 clears the DCBEVT2 event trip condition. Reset type: SYSRSn
5	DCBEVT1	R-0/W1S	0h	Clear Flag for Digital Compare Output B Event 1 0: Writing 0 has no effect. This bit always reads back 0. 1: Writing 1 clears the DCBEVT1 event trip condition. Reset type: SYSRSn
4	DCAEVT2	R-0/W1S	0h	Clear Flag for Digital Compare Output A Event 2 0: Writing 0 has no effect. This bit always reads back 0. 1: Writing 1 clears the DCAEVT2 event trip condition. Reset type: SYSRSn
3	DCAEVT1	R-0/W1S	0h	Clear Flag for Digital Compare Output A Event 1 0: Writing 0 has no effect. This bit always reads back 0. 1: Writing 1 clears the DCAEVT1 event trip condition. Reset type: SYSRSn
2	OST	R-0/W1S	0h	Clear Flag for One-Shot Trip (OST) Latch 0: Has no effect. Always reads back a 0. 1: Clears this Trip (set) condition. Reset type: SYSRSn
1	CBC	R-0/W1S	0h	Clear Flag for Cycle-By-Cycle (CBC) Trip Latch 0: Has no effect. Always reads back a 0. 1: Clears this Trip (set) condition. Reset type: SYSRSn

Table 30-79. TZCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	INT	R-0/W1S	0h	Global Interrupt Clear Flag 0: Has no effect. Always reads back a 0. 1: Clears the trip-interrupt flag for this ePWM module (TZFLG[INT]). NOTE: No further EPWMx_TZINT PIE interrupts will be generated until the flag is cleared. If the TZFLG[INT] bit is cleared and any of the other flag bits are set, then another interrupt pulse will be generated. Clearing all flag bits will prevent further interrupts. Reset type: SYSRSn

30.20.2.54 TZCBCCLR Register (Offset = 130h) [Reset = 0000h]

TZCBCCLR is shown in Figure 30-175 and described in Table 30-80.

Return to the Summary Table.

Trip Zone CBC Clear Register

Figure 30-175. TZCBCCLR Register

15	14	13	12	11	10	9	8
RESERVED							CAPEVT
R-0-0h							R-0/W1S-0h
7	6	5	4	3	2	1	0
DCBEVT2	DCAEVT2	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 30-80. TZCBCCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8	CAPEVT	R-0/W1S	0h	Clear Flag for CAPEVT selected for CBC 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCFLG[CAPEVT] bit. Reset type: SYSRSn
7	DCBEVT2	R-0/W1S	0h	Clear Flag for Digital Compare Output B Event 2 selected for CBC 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCFLG[DCBEVT2] bit. Reset type: SYSRSn
6	DCAEVT2	R-0/W1S	0h	Clear Flag for Digital Compare Output A Event 2 selected for CBC 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCFLG[DCAEVT2] bit. Reset type: SYSRSn
5	CBC6	R-0/W1S	0h	Clear Flag for Cycle-By-Cycle (CBC6) Trip Latch 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCFLG[CBC6] bit. Reset type: SYSRSn
4	CBC5	R-0/W1S	0h	Clear Flag for Cycle-By-Cycle (CBC5) Trip Latch 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCFLG[CBC5] bit. Reset type: SYSRSn
3	CBC4	R-0/W1S	0h	Clear Flag for Cycle-By-Cycle (CBC4) Trip Latch 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCFLG[CBC4] bit. Reset type: SYSRSn
2	CBC3	R-0/W1S	0h	Clear Flag for Cycle-By-Cycle (CBC3) Trip Latch 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCFLG[CBC3] bit. Reset type: SYSRSn
1	CBC2	R-0/W1S	0h	Clear Flag for Cycle-By-Cycle (CBC2) Trip Latch 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCFLG[CBC2] bit. Reset type: SYSRSn
0	CBC1	R-0/W1S	0h	Clear Flag for Cycle-By-Cycle (CBC1) Trip Latch 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCFLG[CBC1] bit. Reset type: SYSRSn

30.20.2.55 TZOSTCLR Register (Offset = 132h) [Reset = 0000h]

TZOSTCLR is shown in Figure 30-176 and described in Table 30-81.

Return to the Summary Table.

Trip Zone OST Clear Register

Figure 30-176. TZOSTCLR Register

15	14	13	12	11	10	9	8
RESERVED							CAPEVT
R-0-0h							R-0/W1S-0h
7	6	5	4	3	2	1	0
DCBEVT1	DCAEVT1	OST6	OST5	OST4	OST3	OST2	OST1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 30-81. TZOSTCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8	CAPEVT	R-0/W1S	0h	Clear Flag for CAPEVT selected for OST 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZOSTFLG[CAPEVT] bit. Reset type: SYSRSn
7	DCBEVT1	R-0/W1S	0h	Clear Flag for Digital Compare Output B Event 1 selected for OST 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZOSTFLG[DCBEVT1] bit. Reset type: SYSRSn
6	DCAEVT1	R-0/W1S	0h	Clear Flag for Digital Compare Output A Event 1 selected for OST 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZOSTFLG[DCAEVT1] bit. Reset type: SYSRSn
5	OST6	R-0/W1S	0h	Clear Flag for Oneshot (OST6) Trip Latch 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZOSTFLG[OST6] bit. Reset type: SYSRSn
4	OST5	R-0/W1S	0h	Clear Flag for Oneshot (OST5) Trip Latch 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZOSTFLG[OST5] bit. Reset type: SYSRSn
3	OST4	R-0/W1S	0h	Clear Flag for Oneshot (OST4) Trip Latch 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZOSTFLG[OST4] bit. Reset type: SYSRSn
2	OST3	R-0/W1S	0h	Clear Flag for Oneshot (OST3) Trip Latch 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZOSTFLG[OST3] bit. Reset type: SYSRSn
1	OST2	R-0/W1S	0h	Clear Flag for Oneshot (OST2) Trip Latch 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZOSTFLG[OST2] bit. Reset type: SYSRSn
0	OST1	R-0/W1S	0h	Clear Flag for Oneshot (OST1) Trip Latch 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZOSTFLG[OST1] bit. Reset type: SYSRSn

30.20.2.56 TZFRC Register (Offset = 136h) [Reset = 0000h]

TZFRC is shown in Figure 30-177 and described in Table 30-82.

Return to the Summary Table.

Trip Zone Force Register

Figure 30-177. TZFRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
CAPEVT	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0-0h

Table 30-82. TZFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R-0	0h	Reserved
7	CAPEVT	R-0/W1S	0h	Force Flag for Capture Event Output 0: Writing 0 has no effect. This bit always reads back 0. 1: Writing 1 forces the CAPEVT event trip condition and sets the TZFLG[CAPEVT] bit. Reset type: SYSRSn
6	DCBEVT2	R-0/W1S	0h	Force Flag for Digital Compare Output B Event 2 0: Writing 0 has no effect. This bit always reads back 0. 1: Writing 1 forces the DCBEVT2 event trip condition and sets the TZFLG[DCBEVT2] bit. Reset type: SYSRSn
5	DCBEVT1	R-0/W1S	0h	Force Flag for Digital Compare Output B Event 1 0: Writing 0 has no effect. This bit always reads back 0. 1: Writing 1 forces the DCBEVT1 event trip condition and sets the TZFLG[DCBEVT1] bit. Reset type: SYSRSn
4	DCAEVT2	R-0/W1S	0h	Force Flag for Digital Compare Output A Event 2 0: Writing 0 has no effect. This bit always reads back 0. 1: Writing 1 forces the DCAEVT2 event trip condition and sets the TZFLG[DCAEVT2] bit. Reset type: SYSRSn
3	DCAEVT1	R-0/W1S	0h	Force Flag for Digital Compare Output A Event 1 0: Writing 0 has no effect. This bit always reads back 0 1: Writing 1 forces the DCAEVT1 event trip condition and sets the TZFLG[DCAEVT1] bit. Reset type: SYSRSn
2	OST	R-0/W1S	0h	Force a One-Shot Trip Event via Software 0: Writing of 0 is ignored. Always reads back a 0. 1: Forces a one-shot trip event and sets the TZFLG[OST] bit. Reset type: SYSRSn
1	CBC	R-0/W1S	0h	Force a Cycle-by-Cycle Trip Event via Software 0: Writing of 0 is ignored. Always reads back a 0. 1: Forces a cycle-by-cycle trip event and sets the TZFLG[CBC] bit. Reset type: SYSRSn
0	RESERVED	R-0	0h	Reserved

30.20.2.57 TZTRIPOUTSEL Register (Offset = 13Ah) [Reset = 0000h]

TZTRIPOUTSEL is shown in Figure 30-178 and described in Table 30-83.

Return to the Summary Table.

Trip Zone Force Register

Figure 30-178. TZTRIPOUTSEL Register

15	14	13	12	11	10	9	8
RESERVED			CAPEVT	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1
R-0-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TZ6	TZ5	TZ4	TZ3	TZ2	TZ1	CBC	OST
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-83. TZTRIPOUTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R-0	0h	Reserved
12	CAPEVT	R/W	0h	CAPEVT Select 0: Disable TZ6 as a TRIPOUT source for this ePWM module 1: Enable TZ6 as a TRIPOUT source for this ePWM module Reset type: SYSRSn
11	DCBEVT2	R/W	0h	DCBEVT2 Select 0: Disable TZ6 as a TRIPOUT source for this ePWM module 1: Enable TZ6 as a TRIPOUT source for this ePWM module Reset type: SYSRSn
10	DCBEVT1	R/W	0h	DCBEVT1 Select 0: Disable TZ6 as a TRIPOUT source for this ePWM module 1: Enable TZ6 as a TRIPOUT source for this ePWM module Reset type: SYSRSn
9	DCAEVT2	R/W	0h	DCAEVT2 Select 0: Disable TZ6 as a TRIPOUT source for this ePWM module 1: Enable TZ6 as a TRIPOUT source for this ePWM module Reset type: SYSRSn
8	DCAEVT1	R/W	0h	DCAEVT1 Select 0: Disable TZ6 as a TRIPOUT source for this ePWM module 1: Enable TZ6 as a TRIPOUT source for this ePWM module Reset type: SYSRSn
7	TZ6	R/W	0h	Trip-zone 6 (TZ6) Select 0: Disable TZ6 as a TRIPOUT source for this ePWM module 1: Enable TZ6 as a TRIPOUT source for this ePWM module Reset type: SYSRSn
6	TZ5	R/W	0h	Trip-zone 5 (TZ5) Select 0: Disable TZ5 as a TRIPOUT source for this ePWM module 1: Enable TZ5 as a TRIPOUT source for this ePWM module Reset type: SYSRSn
5	TZ4	R/W	0h	Trip-zone 4 (TZ4) Select 0: Disable TZ4 as a TRIPOUT source for this ePWM module 1: Enable TZ4 as a TRIPOUT source for this ePWM module Reset type: SYSRSn
4	TZ3	R/W	0h	Trip-zone 3 (TZ3) Select 0: Disable TZ3 as a TRIPOUT source for this ePWM module 1: Enable TZ3 as a TRIPOUT source for this ePWM module Reset type: SYSRSn

Table 30-83. TZTRIPOUTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	TZ2	R/W	0h	Trip-zone 2 (TZ2) Select 0: Disable TZ2 as a TRIPOUT source for this ePWM module 1: Enable TZ2 as a TRIPOUT source for this ePWM module Reset type: SYSRSn
2	TZ1	R/W	0h	Trip-zone 1 (TZ1) Select 0: Disable TZ1 as a TRIPOUT source for this ePWM module 1: Enable TZ1 as a TRIPOUT source for this ePWM module Reset type: SYSRSn
1	CBC	R/W	0h	CBC Select 0: Disable TZ1 as a TRIPOUT source for this ePWM module 1: Enable TZ1 as a TRIPOUT source for this ePWM module Reset type: SYSRSn
0	OST	R/W	0h	OST Select 0: Disable TZ1 as a TRIPOUT source for this ePWM module 1: Enable TZ1 as a TRIPOUT source for this ePWM module Reset type: SYSRSn

30.20.2.58 ETSEL Register (Offset = 148h) [Reset = 0000h]

ETSEL is shown in Figure 30-179 and described in Table 30-84.

Return to the Summary Table.

Event Trigger Selection Register

Figure 30-179. ETSEL Register

15	14	13	12	11	10	9	8
SOCBEN	SOCBSEL			SOCAEN	SOCASEL		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	INTSELCMP	SOCBSELCMP	SOCASELCMP	INTEN	INTSEL		
R-0-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		

Table 30-84. ETSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	SOCBEN	R/W	0h	Enable the ADC Start of Conversion B (EPWMxSOCB) Pulse 0: Disable EPWMxSOCB. 1: Enable EPWMxSOCB pulse. Reset type: SYSRSn
14-12	SOCBSEL	R/W	0h	EPWMxSOCB Selection Options These bits determine when a EPWMxSOCB pulse will be generated. 000: Enable DCBEVT1.soc event 001: Enable event time-base counter equal to zero. (TBCTR = 0x00) 010: Enable event time-base counter equal to period (TBCTR = TBPRD) 011: Enable event time-base counter based on mixed events (ETSOCBMIX). ETSOCBMIX is configured in the ETSOCBMIXEN register. 100: Enable event time-base counter equal to CMPA when the timer is incrementing or CMPC when the timer is incrementing 101: Enable event time-base counter equal to CMPA when the timer is decrementing or CMPC when the timer is decrementing 110: Enable event: time-base counter equal to CMPB when the timer is incrementing or CMPD when the timer is incrementing 111: Enable event: time-base counter equal to CMPB when the timer is decrementing or CMPD when the timer is decrementing (*) Event selected is determined by SOCBSELCMP bit. Reset type: SYSRSn
11	SOCAEN	R/W	0h	Enable the ADC Start of Conversion A (EPWMxSOCA) Pulse 0: Disable EPWMxSOCA. 1: Enable EPWMxSOCA pulse. Reset type: SYSRSn

Table 30-84. ETSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-8	SOCASEL	R/W	0h	EPWMxSOCA Selection Options These bits determine when a EPWMxSOCA pulse will be generated. 000: Enable DCAEVT1.soc event 001: Enable event time-base counter equal to zero. (TBCTR = 0x00) 010: Enable event time-base counter equal to period (TBCTR = TBPRD) 011: Enable event time-base counter based on mixed events (ETSOCAMIX). ETSOCAMIX is configured in the ETSOCAMIXEN register. 100: Enable event time-base counter equal to CMPA when the timer is incrementing or CMPC when the timer is incrementing 101: Enable event time-base counter equal to CMPA when the timer is decrementing or CMPC when the timer is decrementing 110: Enable event: time-base counter equal to CMPB when the timer is incrementing or CMPD when the timer is incrementing 111: Enable event: time-base counter equal to CMPB when the timer is decrementing or CMPD when the timer is decrementing (*) Event selected is determined by SOCASELCMP bit. Reset type: SYSRSn
7	RESERVED	R-0	0h	Reserved
6	INTSELCMP	R/W	0h	EPWMxINT Compare Register Selection Options 0: Enable event time-base counter equal to CMPA when the timer is incrementing / Enable event time-base counter equal to CMPA when the timer is decrementing / Enable event: time-base counter equal to CMPB when the timer is incrementing / Enable event: time-base counter equal to CMPB when the timer is decrementing to INTSEL selection mux. 1: Enable event time-base counter equal to CMPC when the timer is incrementing / Enable event time-base counter equal to CMPC when the timer is decrementing / Enable event: time-base counter equal to CMPD when the timer is incrementing / Enable event: time-base counter equal to CMPD when the timer is decrementing to INTSEL selection mux. Reset type: SYSRSn
5	SOCBSELCMP	R/W	0h	EPWMxSOCB Compare Register Selection Options 0: Enable event time-base counter equal to CMPA when the timer is incrementing / Enable event time-base counter equal to CMPA when the timer is decrementing / Enable event: time-base counter equal to CMPB when the timer is incrementing / Enable event: time-base counter equal to CMPB when the timer is decrementing to SOCBSEL selection mux. 1: Enable event time-base counter equal to CMPC when the timer is incrementing / Enable event time-base counter equal to CMPC when the timer is decrementing / Enable event: time-base counter equal to CMPD when the timer is incrementing / Enable event: time-base counter equal to CMPD when the timer is decrementing to SOCBSEL selection mux. Reset type: SYSRSn
4	SOCASELCMP	R/W	0h	EPWMxSOCA Compare Register Selection Options 0: Enable event time-base counter equal to CMPA when the timer is incrementing / Enable event time-base counter equal to CMPA when the timer is decrementing / Enable event: time-base counter equal to CMPB when the timer is incrementing / Enable event: time-base counter equal to CMPB when the timer is decrementing to SOCASEL selection mux. 1: Enable event time-base counter equal to CMPC when the timer is incrementing / Enable event time-base counter equal to CMPC when the timer is decrementing / Enable event: time-base counter equal to CMPD when the timer is incrementing / Enable event: time-base counter equal to CMPD when the timer is decrementing to SOCASEL selection mux. Reset type: SYSRSn

Table 30-84. ETSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	INTEN	R/W	0h	Enable ePWM Interrupt (EPWMx_INT) Generation 0: Disable EPWMx_INT generation 1: Enable EPWMx_INT generation Reset type: SYSRSn
2-0	INTSEL	R/W	0h	ePWM Interrupt (EPWMx_INT) Selection Options 000: Reserved 001: Enable event time-base counter equal to zero. (TBCTR = 0x00) 010: Enable event time-base counter equal to period (TBCTR = TBPRD) 011: Enable event time-base counter based on mixed events (ETINTMIX). ETINTMIX is configured in the ETINTMIXEN register. 100: Enable event time-base counter equal to CMPA when the timer is incrementing or CMPC when the timer is incrementing 101: Enable event time-base counter equal to CMPA when the timer is decrementing or CMPC when the timer is decrementing 110: Enable event: time-base counter equal to CMPB when the timer is incrementing or CMPD when the timer is incrementing 111: Enable event: time-base counter equal to CMPB when the timer is decrementing or CMPD when the timer is decrementing (*) Event selected is determined by INTSELCMP bit. Reset type: SYSRSn

30.20.2.59 ETPS Register (Offset = 14Ch) [Reset = 0000h]

ETPS is shown in Figure 30-180 and described in Table 30-85.

Return to the Summary Table.

Event Trigger Pre-Scale Register

Figure 30-180. ETPS Register

15	14	13	12	11	10	9	8
SOCBCNT		SOCBPRD		SOCACNT		SOCAPRD	
R-0h		R/W-0h		R-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED		SOCPSSEL	INTPSSEL	INTCNT		INTPRD	
R-0-0h		R/W-0h	R/W-0h	R-0h		R/W-0h	

Table 30-85. ETPS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	SOCBCNT	R	0h	ePWM ADC Start-of-Conversion B Event (EPWMxSOCB) Counter Register These bits indicate how many selected ETSEL[SOCBSEL] events have occurred: 00: No events have occurred. 01: 1 event has occurred. 10: 2 events have occurred. 11: 3 events have occurred. Reset type: SYSRSn
13-12	SOCBPRD	R/W	0h	ePWM ADC Start-of-Conversion B Event (EPWMxSOCB) Period Select These bits determine how many selected ETSEL[SOCBSEL] events need to occur before an EPWMxSOCB pulse is generated. To be generated, the pulse must be enabled (ETSEL[SOCBEN] = 1). The SOCB pulse will be generated even if the status flag is set from a previous start of conversion (ETFLG[SOCB] = 1). Once the SOCB pulse is generated, the ETPS[SOCBCNT] bits will automatically be cleared. 00: Disable the SOCB event counter. No EPWMxSOCB pulse will be generated 01: Generate the EPWMxSOCB pulse on the first event: ETPS[SOCBCNT] = 0,1 10: Generate the EPWMxSOCB pulse on the second event: ETPS[SOCBCNT] = 1,0 11: Generate the EPWMxSOCB pulse on the third event: ETPS[SOCBCNT] = 1,1 Reset type: SYSRSn
11-10	SOCACNT	R	0h	ePWM ADC Start-of-Conversion A Event (EPWMxSOCA) Counter Register These bits indicate how many selected ETSEL[SOCASEL] events have occurred: 00: No events have occurred. 01: 1 event has occurred. 10: 2 events have occurred. 11: 3 events have occurred. Reset type: SYSRSn

Table 30-85. ETPS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9-8	SOCAPRD	R/W	0h	ePWM ADC Start-of-Conversion A Event (EPWMxSOCA) Period Select These bits determine how many selected ETSEL[SOCASEL] events need to occur before an EPWMxSOCA pulse is generated. To be generated, the pulse must be enabled (ETSEL[SOCAEN] = 1). The SOCA pulse will be generated even if the status flag is set from a previous start of conversion (ETFLG[SOCA] = 1). Once the SOCA pulse is generated, the ETPS[SOCACNT] bits will automatically be cleared. 00: Disable the SOCA event counter. No EPWMxSOCA pulse will be generated 01: Generate the EPWMxSOCA pulse on the first event: ETPS[SOCACNT] = 0,1 10: Generate the EPWMxSOCA pulse on the second event: ETPS[SOCACNT] = 1,0 11: Generate the EPWMxSOCA pulse on the third event: ETPS[SOCACNT] = 1,1 Reset type: SYSRSn
7-6	RESERVED	R-0	0h	Reserved
5	SOCPSSEL	R/W	0h	EPWMxSOC A/B Pre-Scale Selection Bits 0: Selects ETPS [SOCACNT/SOCBCNT] and [SOCAPRD/SOCBPRD] registers to determine frequency of events (SOC pulse once every 0-3 events). 1: Selects ETSOCPS [SOCACNT2/SOCBCNT2] and [SOCAPRD2/SOCBPRD2] registers to determine frequency of events (SOC pulse once every 0-15 events). Reset type: SYSRSn
4	INTPSSEL	R/W	0h	EPWMxINTn Pre-Scale Selection Bits 0: Selects ETPS [INTCNT, and INTPRD] registers to determine frequency of events (interrupt once every 0-3 events). 1: Selects ETINTPS [INTCNT2, and INTPRD2] registers to determine frequency of events (interrupt once every 0-15 events). Reset type: SYSRSn
3-2	INTCNT	R	0h	ePWM Interrupt Event (EPWMx_INT) Counter Register These bits indicate how many selected ETSEL[INTSEL] events have occurred. These bits are automatically cleared when an interrupt pulse is generated. If interrupts are disabled, ETSEL[INT] = 0 or the interrupt flag is set, ETFLG[INT] = 1, the counter will stop counting events when it reaches the period value ETPS[INTCNT] = ETPS[INTPRD]. 00: No events have occurred. 01: 1 event has occurred. 10: 2 events have occurred. 11: 3 events have occurred. Reset type: SYSRSn

Table 30-85. ETPS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	INTPRD	R/W	0h	ePWM Interrupt (EPWMx_INT) Period Select These bits determine how many selected ETSEL[INTSEL] events need to occur before an interrupt is generated. To be generated, the interrupt must be enabled (ETSEL[INT] = 1). If the interrupt status flag is set from a previous interrupt (ETFLG[INT] = 1) then no interrupt will be generated until the flag is cleared via the ETCLR[INT] bit. This allows for one interrupt to be pending while another is still being serviced. Once the interrupt is generated, the ETPS[INTCNT] bits will automatically be cleared. Writing a INTPRD value that is the same as the current counter value will trigger an interrupt if it is enabled and the status flag is clear. Writing a INTPRD value that is less than the current counter value will result in an undefined state. If a counter event occurs at the same instant as a new zero or non-zero INTPRD value is written, the counter is incremented. 00: Disable the interrupt event counter. No interrupt will be generated and ETFRC[INT] is ignored. 01: Generate an interrupt on the first event INTCNT = 01 (first event) 10: Generate interrupt on ETPS[INTCNT] = 1,0 (second event) 11: Generate interrupt on ETPS[INTCNT] = 1,1 (third event) Reset type: SYSRSn

30.20.2.60 ETFLG Register (Offset = 150h) [Reset = 0000h]

ETFLG is shown in Figure 30-181 and described in Table 30-86.

Return to the Summary Table.

Event Trigger Flag Register

Figure 30-181. ETFLG Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				SOCB	SOCA	RESERVED	INT
R-0-0h				R-0h	R-0h	R-0-0h	R-0h

Table 30-86. ETFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	SOCB	R	0h	Latched ePWM ADC Start-of-Conversion A (EPWMxSOCB) Status Flag Unlike the ETFLG[INT] flag, the EPWMxSOCB output will continue to pulse even if the flag bit is set. 0: Indicates no event occurred 1: Indicates that a start of conversion pulse was generated on EPWMxSOCB. The EPWMxSOCB output will continue to be generated even if the flag bit is set. Reset type: SYSRSn
2	SOCA	R	0h	Latched ePWM ADC Start-of-Conversion A (EPWMxSOCA) Status Flag Unlike the ETFLG[INT] flag, the EPWMxSOCA output will continue to pulse even if the flag bit is set. 0: Indicates no event occurred 1: Indicates that a start of conversion pulse was generated on EPWMxSOCA. The EPWMxSOCA output will continue to be generated even if the flag bit is set. Reset type: SYSRSn
1	RESERVED	R-0	0h	Reserved
0	INT	R	0h	Latched ePWM Interrupt (EPWMx_INT) Status Flag 0: Indicates no event occurred 1: Indicates that an ePWMx interrupt (EPWMx_INT) was generated. No further interrupts will be generated until the flag bit is cleared. Up to one interrupt can be pending while the ETFLG[INT] bit is still set. If an interrupt is pending, it will not be generated until after the ETFLG[INT] bit is cleared. Reset type: SYSRSn

30.20.2.61 ETCLR Register (Offset = 154h) [Reset = 0000h]

ETCLR is shown in Figure 30-182 and described in Table 30-87.

Return to the Summary Table.

Event Trigger Clear Register

Figure 30-182. ETCLR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				SOCB	SOCA	RESERVED	INT
R-0-0h				R-0/W1S-0h	R-0/W1S-0h	R-0-0h	R-0/W1S-0h

Table 30-87. ETCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	SOCB	R-0/W1S	0h	ePWM ADC Start-of-Conversion A (EPWMxSOCB) Flag Clear Bit 0: Writing a 0 has no effect. Always reads back a 0 1: Clears the ETFLG[SOCB] flag bit Reset type: SYSRSn
2	SOCA	R-0/W1S	0h	ePWM ADC Start-of-Conversion A (EPWMxSOCA) Flag Clear Bit 0: Writing a 0 has no effect. Always reads back a 0 1: Clears the ETFLG[SOCA] flag bit Reset type: SYSRSn
1	RESERVED	R-0	0h	Reserved
0	INT	R-0/W1S	0h	ePWM Interrupt (EPWMx_INT) Flag Clear Bit 0: Writing a 0 has no effect. Always reads back a 0 1: Clears the ETFLG[INT] flag bit and enable further interrupts pulses to be generated Reset type: SYSRSn

30.20.2.62 ETFRC Register (Offset = 158h) [Reset = 0000h]

ETFRC is shown in Figure 30-183 and described in Table 30-88.

Return to the Summary Table.

Event Trigger Force Register

Figure 30-183. ETFRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				SOCB	SOCA	RESERVED	INT
R-0-0h				R-0/W1S-0h	R-0/W1S-0h	R-0-0h	R-0/W1S-0h

Table 30-88. ETFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	SOCB	R-0/W1S	0h	SOCB Force Bit The SOCB pulse will only be generated if the event is enabled in the ETSEL register. The ETFLG[SOCB] flag bit will be set regardless. 0: Writing 0 to this bit will be ignored. Always reads back a 0. 1: Generates a pulse on EPWMxSOCB and set the SOCBFLG bit. This bit is used for test purposes. Reset type: SYSRSn
2	SOCA	R-0/W1S	0h	SOCA Force Bit The SOCA pulse will only be generated if the event is enabled in the ETSEL register. The ETFLG[SOCA] flag bit will be set regardless. 0: Writing 0 to this bit will be ignored. Always reads back a 0. 1: Generates a pulse on EPWMxSOCA and set the SOCAFLG bit. This bit is used for test purposes. Reset type: SYSRSn
1	RESERVED	R-0	0h	Reserved
0	INT	R-0/W1S	0h	INT Force Bit The interrupt will only be generated if the event is enabled in the ETSEL register. The INT flag bit will be set regardless. 0: Writing 0 to this bit will be ignored. Always reads back a 0. 1: Generates an interrupt on EPWMxINT and set the INT flag bit. This bit is used for test purposes. Reset type: SYSRSn

30.20.2.63 ETINTPS Register (Offset = 15Ch) [Reset = 0000h]

ETINTPS is shown in Figure 30-184 and described in Table 30-89.

Return to the Summary Table.

Event-Trigger Interrupt Pre-Scale Register

Figure 30-184. ETINTPS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
INTCNT2				INTPRD2			
R-0h				R/W-0h			

Table 30-89. ETINTPS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R-0	0h	Reserved
7-4	INTCNT2	R	0h	EPWMxINT Counter 2 When ETPS[INTPSSEL]=1, these bits indicate how many selected events have occurred: 0000: No events 0001: 1 event 0010: 2 events 0011: 3 events 0100: 4 events ... 1111: 15 events Reset type: SYSRSn
3-0	INTPRD2	R/W	0h	EPWMxINT Period 2 Select When ETPS[INTPSSEL] = 1, these bits select how many selected events need to occur before an interrupt is generated: 0000: Disable counter 0001: Generate interrupt on INTCNT = 1 (first event) 0010: Generate interrupt on INTCNT = 2 (second event) 0011: Generate interrupt on INTCNT = 3 (third event) 0100: Generate interrupt on INTCNT = 4 (fourth event) ... 1111: Generate interrupt on INTCNT = 15 (fifteenth event) Reset type: SYSRSn

30.20.2.64 ETSOCPS Register (Offset = 160h) [Reset = 0000h]

ETSOCPS is shown in Figure 30-185 and described in Table 30-90.

Return to the Summary Table.

Event-Trigger SOC Pre-Scale Register

Figure 30-185. ETSOCPS Register

15	14	13	12	11	10	9	8
SOCBCNT2				SOCBPRD2			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
SOCACNT2				SOCAPRD2			
R-0h				R/W-0h			

Table 30-90. ETSOCPS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	SOCBCNT2	R	0h	EPWMxSOCB Counter 2 When ETPS[SOCPSSEL] = 1, these bits indicate how many selected events have occurred: 0000: No events 0001: 1 event 0010: 2 events 0011: 3 events 0100: 4 events ... 1111: 15 events Reset type: SYSRSn
11-8	SOCBPRD2	R/W	0h	EPWMxSOCB Period 2 Select When ETPS[SOCPSSEL] = 1, these bits select how many selected event need to occur before an SOCB pulse is generated: 0000: Disable counter 0001: Generate SOC pulse on SOCBCNT2 = 1 (first event) 0010: Generate SOC pulse on SOCBCNT2 = 2 (second event) 0011: Generate SOC pulse on SOCBCNT2 = 3 (third event) 0100: Generate SOC pulse on SOCBCNT2 = 4 (fourth event) ... 1111: Generate SOC pulse on SOCBCNT2 = 15 (fifteenth event) Reset type: SYSRSn
7-4	SOCACNT2	R	0h	EPWMxSOCA Counter 2 When ETPS[SOCPSSEL] = 1, these bits indicate how many selected events have occurred: 0000: No events 0001: 1 event 0010: 2 events 0011: 3 events 0100: 4 events ... 1111: 15 events Reset type: SYSRSn

Table 30-90. ETSOCPS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	SOCAPRD2	R/W	0h	EPWMxSOCA Period 2 Select When ETPS[SOCPSSEL] = 1, these bits select how many selected event need to occur before an SOCA pulse is generated: 0000: Disable counter 0001: Generate SOC pulse on SOCACNT2 = 1 (first event) 0010: Generate SOC pulse on SOCACNT2 = 2 (second event) 0011: Generate SOC pulse on SOCACNT2 = 3 (third event) 0100: Generate SOC pulse on SOCACNT2 = 4 (fourth event) ... 1111: Generate SOC pulse on SOCACNT2 = 15 (fifteenth event) Reset type: SYSRSn

30.20.2.65 ETCNTINITCTL Register (Offset = 164h) [Reset = 0000h]

ETCNTINITCTL is shown in Figure 30-186 and described in Table 30-91.

Return to the Summary Table.

Event-Trigger Counter Initialization Control Register

Figure 30-186. ETCNTINITCTL Register

15	14	13	12	11	10	9	8
SOCBINITEN	SOCAINITEN	INTINITEN	SOCBINITFRC	SOCAINITFRC	INTINITFRC	RESERVED	
R/W-0h	R/W-0h	R/W-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0-0h	
7	6	5	4	3	2	1	0
RESERVED							
R-0-0h							

Table 30-91. ETCNTINITCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	SOCBINITEN	R/W	0h	EPWMxSOCB Counter 2 Initialization Enable 0: Has no effect. 1: Enable initialization of EPWMxSOCB counter with contents of ETCNTINIT[SOCBINIT] on a SYNC event or software force. Reset type: SYSRSn
14	SOCAINITEN	R/W	0h	EPWMxSOCA Counter 2 Initialization Enable 0: Has no effect. 1: Enable initialization of EPWMxSOCA counter with contents of ETCNTINIT[SOCAINIT] on a SYNC event or software force. Reset type: SYSRSn
13	INTINITEN	R/W	0h	EPWMxINT Counter 2 Initialization Enable 0: Has no effect. 1: Enable initialization of EPWMxINT counter 2 with contents of ETCNTINIT[INTINIT] on a SYNC event or software force. Reset type: SYSRSn
12	SOCBINITFRC	R-0/W1S	0h	EPWMxSOCB Counter 2 Initialization Force 0: Has no effect. 1: This bit forces the ET EPWMxSOCB counter to be initialized with the contents of ETCNTINIT[SOCBINIT]. Reset type: SYSRSn
11	SOCAINITFRC	R-0/W1S	0h	EPWMxSOCA Counter 2 Initialization Force 0: Has no effect. 1: This bit forces the ET EPWMxSOCA counter to be initialized with the contents of ETCNTINIT[SOCAINIT]. Reset type: SYSRSn
10	INTINITFRC	R-0/W1S	0h	EPWMxINT Counter 2 Initialization Force 0: Has no effect. 1: This bit forces the ET EPWMxINT counter to be initialized with the contents of ETCNTINIT[INTINIT]. Reset type: SYSRSn
9-0	RESERVED	R-0	0h	Reserved

30.20.2.66 ETCNTINIT Register (Offset = 168h) [Reset = 0000h]

ETCNTINIT is shown in Figure 30-187 and described in Table 30-92.

Return to the Summary Table.

Event-Trigger Counter Initialization Register

Figure 30-187. ETCNTINIT Register

15	14	13	12	11	10	9	8
RESERVED				SOCBINIT			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
SOCAINIT				INTINIT			
R/W-0h				R/W-0h			

Table 30-92. ETCNTINIT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-8	SOCBINIT	R/W	0h	EPWMxSOCB Counter 2 Initialization Bits The ET EPWMxSOCB counter is initialized with the contents of this register on an ePWM SYNC event or a software force. Reset type: SYSRSn
7-4	SOCAINIT	R/W	0h	EPWMxSOCA Counter 2 Initialization Bits The ET EPWMxSOCA counter is initialized with the contents of this register on an ePWM SYNC event or a software force. Reset type: SYSRSn
3-0	INTINIT	R/W	0h	EPWMxINT Counter 2 Initialization Bits The ET EPWMxINT counter is initialized with the contents of this register on an ePWM SYNC event or a software force. Reset type: SYSRSn

30.20.2.67 ETINTMIXEN Register (Offset = 16Ch) [Reset = 0003h]

ETINTMIXEN is shown in Figure 30-188 and described in Table 30-93.

Return to the Summary Table.

Event-Trigger Mixed INT Selection

Figure 30-188. ETINTMIXEN Register

15	14	13	12	11	10	9	8
RESERVED					DCAEVT1	CDD	CDU
R-0-0h					R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CCD	CCU	CBD	CBU	CAD	CAU	PRD	ZRO
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h

Table 30-93. ETINTMIXEN Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R-0	0h	Reserved
10	DCAEVT1	R/W	0h	Enable DCAEVT1.inter to the mixed ET interrupt trigger signal (ETINTMIX). 0: DCAEVT1.soc event is not enabled 1: Enable DCAEVT1.soc event Reset type: SYSRSn
9	CDD	R/W	0h	Enable event time-base counter equal to CMPD when the timer is decrementing to the mixed ET interrupt trigger signal (ETINTMIX). 0: CMPD down-count match enable event is not enabled 1: Enable CMPD down-count match enable event Reset type: SYSRSn
8	CDU	R/W	0h	Enable event time-base counter equal to CMPD when the timer is incrementing to the mixed ET interrupt trigger signal (ETINTMIX). 0: CMPD up-count match enable event is not enabled 1: Enable CMPD up-count match enable event Reset type: SYSRSn
7	CCD	R/W	0h	Enable event time-base counter equal to CMPC when the timer is decrementing to the mixed ET interrupt trigger signal (ETINTMIX). 0: CMPC down-count match enable event is not enabled 1: Enable CMPC down-count match enable event Reset type: SYSRSn
6	CCU	R/W	0h	Enable event time-base counter equal to CMPC when the timer is incrementing to the mixed ET interrupt trigger signal (ETINTMIX). 0: CMPC up-count match enable event is not enabled 1: Enable CMPC up-count match enable event Reset type: SYSRSn
5	CBD	R/W	0h	Enable event time-base counter equal to CMPB when the timer is decrementing to the mixed ET interrupt trigger signal (ETINTMIX). 0: CMPB down-count match enable event is not enabled 1: Enable CMPB down-count match enable event Reset type: SYSRSn
4	CBU	R/W	0h	Enable event time-base counter equal to CMPB when the timer is incrementing to the mixed ET interrupt trigger signal (ETINTMIX). 0: CMPB up-count match enable event is not enabled 1: Enable CMPB up-count match enable event Reset type: SYSRSn

Table 30-93. ETINTMIXEN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	CAD	R/W	0h	Enable event time-base counter equal to CMPA when the timer is decrementing to the mixed ET interrupt trigger signal (ETINTMIX). 0: CMPA down-count match enable event is not enabled 1: Enable CMPA down-count match enable event Reset type: SYSRSn
2	CAU	R/W	0h	Enable event time-base counter equal to CMPA when the timer is incrementing to the mixed ET interrupt trigger signal (ETINTMIX). 0: CMPA up-count match enable event is not enabled 1: Enable CMPA up-count match enable event Reset type: SYSRSn
1	PRD	R/W	1h	Enable event time-base counter equal to period (TBCTR = TBPRD) to the mixed ET interrupt trigger signal (ETINTMIX). 0: Period match event is not enabled 1: Enable period match event Reset type: SYSRSn
0	ZRO	R/W	1h	Enable event time-base counter equal to zero (TBCTR = 0x00) to the mixed ET interrupt trigger signal (ETINTMIX). 0: Zero match event is not enabled 1: Enable zero match event Reset type: SYSRSn

30.20.2.68 ETSOCAMIXEN Register (Offset = 170h) [Reset = 0003h]

ETSOCAMIXEN is shown in Figure 30-189 and described in Table 30-94.

Return to the Summary Table.

Event-Trigger Mixed SOCA Selection

Figure 30-189. ETSOCAMIXEN Register

15	14	13	12	11	10	9	8
RESERVED					DCAEVT1	CDD	CDU
R-0-0h					R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CCD	CCU	CBD	CBU	CAD	CAU	PRD	ZRO
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h

Table 30-94. ETSOCAMIXEN Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R-0	0h	Reserved
10	DCAEVT1	R/W	0h	Enable DCAEVT1.inter to the mixed ET SOCA trigger signal (ETSOCAMIX). 0: DCAEVT1.soc event is not enabled 1: Enable DCAEVT1.soc event Reset type: SYSRSn
9	CDD	R/W	0h	Enable event time-base counter equal to CMPD when the timer is decrementing to the mixed ET SOCA trigger signal (ETSOCAMIX). 0: CMPD down-count match enable event is not enabled 1: Enable CMPD down-count match enable event Reset type: SYSRSn
8	CDU	R/W	0h	Enable event time-base counter equal to CMPD when the timer is incrementing to the mixed ET SOCA trigger signal (ETSOCAMIX). 0: CMPD up-count match enable event is not enabled 1: Enable CMPD up-count match enable event Reset type: SYSRSn
7	CCD	R/W	0h	Enable event time-base counter equal to CMPC when the timer is decrementing to the mixed ET SOCA trigger signal (ETSOCAMIX). 0: CMPC down-count match enable event is not enabled 1: Enable CMPC down-count match enable event Reset type: SYSRSn
6	CCU	R/W	0h	Enable event time-base counter equal to CMPC when the timer is incrementing to the mixed ET SOCA trigger signal (ETSOCAMIX). 0: CMPC up-count match enable event is not enabled 1: Enable CMPC up-count match enable event Reset type: SYSRSn
5	CBD	R/W	0h	Enable event time-base counter equal to CMPB when the timer is decrementing to the mixed ET SOCA trigger signal (ETSOCAMIX). 0: CMPB down-count match enable event is not enabled 1: Enable CMPB down-count match enable event Reset type: SYSRSn
4	CBU	R/W	0h	Enable event time-base counter equal to CMPB when the timer is incrementing to the mixed ET SOCA trigger signal (ETSOCAMIX). 0: CMPB up-count match enable event is not enabled 1: Enable CMPB up-count match enable event Reset type: SYSRSn

Table 30-94. ETSOCAMIXEN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	CAD	R/W	0h	Enable event time-base counter equal to CMPA when the timer is decrementing to the mixed ET SOCA trigger signal (ETSOCAMIX). 0: CMPA down-count match enable event is not enabled 1: Enable CMPA down-count match enable event Reset type: SYSRSn
2	CAU	R/W	0h	Enable event time-base counter equal to CMPA when the timer is incrementing to the mixed ET SOCA trigger signal (ETSOCAMIX). 0: CMPA up-count match enable event is not enabled 1: Enable CMPA up-count match enable event Reset type: SYSRSn
1	PRD	R/W	1h	Enable event time-base counter equal to period (TBCTR = TBPRD) to the mixed ET SOCA trigger signal (ETSOCAMIX). 0: Period match event is not enabled 1: Enable period match event Reset type: SYSRSn
0	ZRO	R/W	1h	Enable event time-base counter equal to zero (TBCTR = 0x00) to the mixed ET SOCA trigger signal (ETSOCAMIX). 0: Zero match event is not enabled 1: Enable zero match event Reset type: SYSRSn

30.20.2.69 ETSOCBMIXEN Register (Offset = 174h) [Reset = 0003h]

ETSOCBMIXEN is shown in Figure 30-190 and described in Table 30-95.

Return to the Summary Table.

Event-Trigger Mixed SOCB Selection

Figure 30-190. ETSOCBMIXEN Register

15	14	13	12	11	10	9	8
RESERVED					DCBEVT1	CDD	CDU
R-0-0h					R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CCD	CCU	CBD	CBU	CAD	CAU	PRD	ZRO
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h

Table 30-95. ETSOCBMIXEN Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R-0	0h	Reserved
10	DCBEVT1	R/W	0h	Enable DCBEVT1.inter to the mixed ET SOCB trigger signal (ETSOCBMIX). 0: DCBEVT1.soc event is not enabled 1: Enable DCBEVT1.soc event Reset type: SYSRSn
9	CDD	R/W	0h	Enable event time-base counter equal to CMPD when the timer is decrementing to the mixed ET SOCB trigger signal (ETSOCBMIX). 0: CMPD down-count match enable event is not enabled 1: Enable CMPD down-count match enable event Reset type: SYSRSn
8	CDU	R/W	0h	Enable event time-base counter equal to CMPD when the timer is incrementing to the mixed ET SOCB trigger signal (ETSOCBMIX). 0: CMPD up-count match enable event is not enabled 1: Enable CMPD up-count match enable event Reset type: SYSRSn
7	CCD	R/W	0h	Enable event time-base counter equal to CMPC when the timer is decrementing to the mixed ET SOCB trigger signal (ETSOCBMIX). 0: CMPC down-count match enable event is not enabled 1: Enable CMPC down-count match enable event Reset type: SYSRSn
6	CCU	R/W	0h	Enable event time-base counter equal to CMPC when the timer is incrementing to the mixed ET SOCB trigger signal (ETSOCBMIX). 0: CMPC up-count match enable event is not enabled 1: Enable CMPC up-count match enable event Reset type: SYSRSn
5	CBD	R/W	0h	Enable event time-base counter equal to CMPB when the timer is decrementing to the mixed ET SOCB trigger signal (ETSOCBMIX). 0: CMPB down-count match enable event is not enabled 1: Enable CMPB down-count match enable event Reset type: SYSRSn
4	CBU	R/W	0h	Enable event time-base counter equal to CMPB when the timer is incrementing to the mixed ET SOCB trigger signal (ETSOCBMIX). 0: CMPB up-count match enable event is not enabled 1: Enable CMPB up-count match enable event Reset type: SYSRSn

Table 30-95. ETSOCBMIXEN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	CAD	R/W	0h	Enable event time-base counter equal to CMPA when the timer is decrementing to the mixed ET SOCB trigger signal (ETSOCBMIX). 0: CMPA down-count match enable event is not enabled 1: Enable CMPA down-count match enable event Reset type: SYSRSn
2	CAU	R/W	0h	Enable event time-base counter equal to CMPA when the timer is incrementing to the mixed ET SOCB trigger signal (ETSOCBMIX). 0: CMPA up-count match enable event is not enabled 1: Enable CMPA up-count match enable event Reset type: SYSRSn
1	PRD	R/W	1h	Enable event time-base counter equal to period (TBCTR = TBPRD) to the mixed ET SOCB trigger signal (ETSOCBMIX). 0: Period match event is not enabled 1: Enable period match event Reset type: SYSRSn
0	ZRO	R/W	1h	Enable event time-base counter equal to zero (TBCTR = 0x00) to the mixed ET SOCB trigger signal (ETSOCBMIX). 0: Zero match event is not enabled 1: Enable zero match event Reset type: SYSRSn

30.20.2.70 DCTRIPSEL Register (Offset = 180h) [Reset = 0000h]

DCTRIPSEL is shown in Figure 30-191 and described in Table 30-96.

Return to the Summary Table.

Digital Compare Trip Select Register

Figure 30-191. DCTRIPSEL Register

15	14	13	12	11	10	9	8
DCBLCOMPSEL				DCBHCOMPSEL			
R/W-0h				R/W-0h			
7	6	5	4	3	2	1	0
DCALCOMPSEL				DCAHCOMPSEL			
R/W-0h				R/W-0h			

Table 30-96. DCTRIPSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	DCBLCOMPSEL	R/W	0h	Digital Compare B Low Input Select Bits 0000: TRIPIN1 0001: TRIPIN2 0010: TRIPIN3 0011: TRIPIN4 ... 1011: TRIPIN12 1100: Reserved 1101: TRIPIN14 1110: TRIPIN15 1111: Trip combination input (all trip inputs selected by DCBLTRIPSEL register ORed together) Reset type: SYSRSn
11-8	DCBHCOMPSEL	R/W	0h	Digital Compare B High Input Select Bits 0000: TRIPIN1 0001: TRIPIN2 0010: TRIPIN3 0011: TRIPIN4 ... 1011: TRIPIN12 1100: Reserved 1101: TRIPIN14 1110: TRIPIN15 1111: Trip combination input (all trip inputs selected by DCBHTRIPSEL register ORed together) Reset type: SYSRSn
7-4	DCALCOMPSEL	R/W	0h	Digital Compare A Low Input Select Bits 0000: TRIPIN1 0001: TRIPIN2 0010: TRIPIN3 0011: TRIPIN4 ... 1011: TRIPIN12 1100: Reserved 1101: TRIPIN14 1110: TRIPIN15 1111: Trip combination input (all trip inputs selected by DCALTRIPSEL register ORed together) Reset type: SYSRSn

Table 30-96. DCTRIPSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	DCAHCOMPSEL	R/W	0h	Digital Compare A High Input Select Bits 0000: TRIPIN1 0001: TRIPIN2 0010: TRIPIN3 0011: TRIPIN4 ... 1011: TRIPIN12 1100: Reserved 1101: TRIPIN14 1110: TRIPIN15 1111: Trip combination input (all trip inputs selected by DCAHTRIPSEL register ORed together) Reset type: SYSRSn

30.20.2.71 DCACTL Register (Offset = 186h) [Reset = 0000h]

DCACTL is shown in Figure 30-192 and described in Table 30-97.

Return to the Summary Table.

Digital Compare A Control Register

Figure 30-192. DCACTL Register

15	14	13	12	11	10	9	8
EVT2LAT	EVT2LATCLRSEL		EVT2LATSEL	RESERVED		EVT2FRCSYNCSEL	EVT2SRCSEL
R-0h	R/W-0h		R/W-0h	R-0-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
EVT1LAT	EVT1LATCLRSEL		EVT1LATSEL	EVT1SYNCE	EVT1SOCE	EVT1FRCSYNCSEL	EVT1SRCSEL
R-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-97. DCACTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	EVT2LAT	R	0h	Indicates the status of DCAEVT2LAT signal. 0 : The DCAEVT2LAT latch is cleared. 1 : The DCAEVT2LAT latch is set. Reset type: SYSRSn
14-13	EVT2LATCLRSEL	R/W	0h	DCAEVT2 Latched clear source select: 00: CNT_ZERO event clears DCAEVT2 latch. 01: PRD_EQ event clears DCAEVT2 latch. 10: CNT_ZERO event or PRD_EQ event clears DCAEVT2 latch. 11: Reserved. Reset type: SYSRSn
12	EVT2LATSEL	R/W	0h	DCAEVT2 Latched signal select: 0: Does not select the DCAEVT2 latched signal as source of DCAEVT2.force. 1: Selects the DCAEVT2 latched signal as source of DCAEVT2.force. Reset type: SYSRSn
11-10	RESERVED	R-0	0h	Reserved
9	EVT2FRCSYNCSEL	R/W	0h	DCAEVT2 Force Synchronization Signal Select 0: Source is synchronized with EPWMCLK 1: Source is passed through asynchronously Reset type: SYSRSn
8	EVT2SRCSEL	R/W	0h	DCAEVT2 Source Signal Select 0: Source Is DCAEVT2 Signal 1: Source Is DCEVTFILT Signal Reset type: SYSRSn
7	EVT1LAT	R	0h	Indicates the status of DCAEVT1LAT signal. 0 : The DCAEVT1LAT latch is cleared. 1 : The DCAEVT1LAT latch is set. Reset type: SYSRSn
6-5	EVT1LATCLRSEL	R/W	0h	DCAEVT1 Latched clear source select: 00: CNT_ZERO event clears DCAEVT1 latch. 01: PRD_EQ event clears DCAEVT1 latch. 10: CNT_ZERO event or PRD_EQ event clears DCAEVT1 latch. 11 : Reserved. Reset type: SYSRSn

Table 30-97. DCACTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	EVT1LATSEL	R/W	0h	DCAEVT1 Latched signal select: 0: Does not select the DCAEVT1 latched signal as source of DCAEVT1.force. 1: Selects the DCAEVT1 latched signal as source of DCAEVT1.force. Reset type: SYSRSn
3	EVT1SYNCE	R/W	0h	DCAEVT1 SYNC, Enable/Disable 0: SYNC Generation Disabled 1: SYNC Generation Enabled Reset type: SYSRSn
2	EVT1SOCE	R/W	0h	DCAEVT1 SOC, Enable/Disable 0: SOC Generation Disabled 1: SOC Generation Enabled Reset type: SYSRSn
1	EVT1FRCSYNCSEL	R/W	0h	DCAEVT1 Force Synchronization Signal Select 0: Source is synchronized with EPWMCLK 1: Source is passed through asynchronously Reset type: SYSRSn
0	EVT1SRCSEL	R/W	0h	DCAEVT1 Source Signal Select 0: Source Is DCAEVT1 Signal 1: Source Is DCEVTFILT Signal Reset type: SYSRSn

30.20.2.72 DCBCTL Register (Offset = 188h) [Reset = 0000h]

DCBCTL is shown in Figure 30-193 and described in Table 30-98.

Return to the Summary Table.

Digital Compare B Control Register

Figure 30-193. DCBCTL Register

15	14	13	12	11	10	9	8
EVT2LAT	EVT2LATCLRSEL		EVT2LATSEL	RESERVED		EVT2FRCSYNCSEL	EVT2SRCSEL
R-0h	R/W-0h		R/W-0h	R-0-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
EVT1LAT	EVT1LATCLRSEL		EVT1LATSEL	EVT1SYNCE	EVT1SOCE	EVT1FRCSYNCSEL	EVT1SRCSEL
R-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-98. DCBCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	EVT2LAT	R	0h	Indicates the status of DCBEVT2LAT signal. 0 The DCBEVT2LAT latch is cleared. 1 The DCBEVT2LAT latch is set. Reset type: SYSRSn
14-13	EVT2LATCLRSEL	R/W	0h	DCBEVT2 Latched clear source select: 00 CNT_ZERO event clears DCBEVT2 latch. 01 PRD_EQ event clears DCBEVT2 latch. 10 CNT_ZERO event or PRD_EQ event clears DCBEVT2 latch. 11 Reserved. Reset type: SYSRSn
12	EVT2LATSEL	R/W	0h	DCBEVT2 Latched signal select: 0 Does not select the DCBEVT2 latched signal (Refer figure 'Modifications to DCBEVT1.force/DCBEVT2.force generation.') as source of DCBEVT2.force. 1 Selects the DCBEVT2 latched signal as source of DCBEVT2.force. Reset type: SYSRSn
11-10	RESERVED	R-0	0h	Reserved
9	EVT2FRCSYNCSEL	R/W	0h	DCBEVT2 Force Synchronization Signal Select 0: Source is synchronized with EPWMCLK 1: Source is passed through asynchronously Reset type: SYSRSn
8	EVT2SRCSEL	R/W	0h	DCBEVT2 Source Signal Select 0: Source Is DCBEVT2 Signal 1: Source Is DCEVTFILT Signal Reset type: SYSRSn
7	EVT1LAT	R	0h	Indicates the status of DCBEVT1LAT signal. 0 The DCBEVT1LAT latch is cleared. 1 The DCBEVT1LAT latch is set. Reset type: SYSRSn
6-5	EVT1LATCLRSEL	R/W	0h	DCBEVT1 Latched clear source select: 00 CNT_ZERO event clears DCBEVT1 latch. 01 PRD_EQ event clears DCBEVT1 latch. 10 CNT_ZERO event or PRD_EQ event clears DCBEVT1 latch. 11 Reserved. Reset type: SYSRSn

Table 30-98. DCBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	EVT1LATSEL	R/W	0h	DCBEVT1 Latched signal select: 0 Does not select the DCBEVT1 latched signal (Refer figure 'Modifications to DCBEVT1.force/DCBEVT2.force generation.') as source of DCBEVT1.force. 1 Selects the DCBEVT1 latched signal as source of DCBEVT1.force. Reset type: SYSRSn
3	EVT1SYNCE	R/W	0h	DCBEVT1 SYNC, Enable/Disable 0: SYNC Generation Disabled 1: SYNC Generation Enabled Reset type: SYSRSn
2	EVT1SOCE	R/W	0h	DCBEVT1 SOC, Enable/Disable 0: SOC Generation Disabled 1: SOC Generation Enabled Reset type: SYSRSn
1	EVT1FRCSYNCSEL	R/W	0h	DCBEVT1 Force Synchronization Signal Select 0: Source is synchronized with EPWMCLK 1: Source is passed through asynchronously Reset type: SYSRSn
0	EVT1SRCSEL	R/W	0h	DCBEVT1 Source Signal Select 0: Source Is DCBEVT1 Signal 1: Source Is DCEVTFILT Signal Reset type: SYSRSn

30.20.2.73 DCFCTL Register (Offset = 18Eh) [Reset = 0000h]

DCFCTL is shown in Figure 30-194 and described in Table 30-99.

Return to the Summary Table.

Digital Compare Filter Control Register

Figure 30-194. DCFCTL Register

15	14	13	12	11	10	9	8
EDGESTATUS			EDGECOUNT			EDGEMODE	
R-0h			R/W-0h			R/W-0h	
7	6	5	4	3	2	1	0
RESERVED	EDGEFILTSEL	PULSESEL		BLANKINV	BLANKE	SRCSEL	
R-0-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	

Table 30-99. DCFCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	EDGESTATUS	R	0h	Edge Status: These bits reflect the total number of edges currently captured. When the value matches the EDGECOUNT, the status bits are set to zero. and a TBCLK wide pulse is generated which can then be output on the DCEVTFILT signal. The edge counter can be reset by writing 000 to the EDGECOUNT value: Reset type: SYSRSn
12-10	EDGECOUNT	R/W	0h	Edge Count: These bits select how many edges to count before generating a TBCLK wide pulse on the DCEVTFILT signal: 000: no edges, reset current EDGESTATUS bits to 0,0,0 001: 1 edge 010: 2 edges 011: 3 edges 100: 4 edges 101: 5 edges 110: 6 edges 111: 7 edges Reset type: SYSRSn
9-8	EDGEMODE	R/W	0h	Edge Mode Select: 00: Low To High Edge 01: High To Low Edge 10: Both Edges 11: Reserved Reset type: SYSRSn
7	RESERVED	R-0	0h	Reserved
6	EDGEFILTSEL	R/W	0h	Edge Filter Select: 0: Edge Filter Not Selected 1: Edge Filter Selected Reset type: SYSRSn
5-4	PULSESEL	R/W	0h	Pulse Select For Blanking & Capture Alignment 00: Time-base counter equal to period (TBCTR = TBPRD) 01: Time-base counter equal to zero (TBCTR = 0x00) 10: Time-base counter equal to zero (TBCTR = 0x00) or period (TBCTR = TBPRD) 11: BLANKPULSEMIX Reset type: SYSRSn
3	BLANKINV	R/W	0h	Blanking Window Inversion 0: Blanking window not inverted 1: Blanking window inverted Reset type: SYSRSn

Table 30-99. DCFCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	BLANKE	R/W	0h	Blanking Window Enable/Disable 0: Blanking window is disabled 1: Blanking window is enabled Reset type: SYSRSn
1-0	SRCSEL	R/W	0h	Filter Block Signal Source Select 00: Source Is DCAEVT1 Signal 01: Source Is DCAEVT2 Signal 10: Source Is DCBEVT1 Signal 11: Source Is DCBEVT2 Signal Reset type: SYSRSn

30.20.2.74 DCCAPCTL Register (Offset = 190h) [Reset = 0000h]

DCCAPCTL is shown in Figure 30-195 and described in Table 30-100.

Return to the Summary Table.

Digital Compare Capture Control Register

Figure 30-195. DCCAPCTL Register

15	14	13	12	11	10	9	8
CAPMODE	CAPCLR	CAPSTS	RESERVED				
R/W-0h	R-0/W1S-0h	R-0h	R-0-0h				
7	6	5	4	3	2	1	0
RESERVED						SHDWMODE	CAPE
R-0-0h						R/W-0h	R/W-0h

Table 30-100. DCCAPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	CAPMODE	R/W	0h	Counter Capture Mode 0: When a DCEVTFILT occurs and the counter capture is enabled, then the current TBCNT value is captured in the active register. When the respective trip event occurs, further trip (capture) events are ignored until the next PRD_eq or CNT_zero event (as selected by the PULSESEL bit in the DCFCTL register) re-triggers the capture mechanism. If active mode is enabled, via SHDWMODE bit in DCCAPCTL register, CPU reads of this register will return the active register value. If shadow mode is enabled, via SHDWMODE bit in DCCAPCTL register, the active register is copied to the shadow register on the PRD_eq or CNT_zero event (whichever is selected by PULSESEL bit in DCFCTL register). CPU reads of this register will return the shadow register value. 1: When a DCEVTFILT occurs and the counter capture is enabled, then the current TBCNT value is captured in the active register. When the respective trip event occurs - it will set the CAPSTS flag and further trip (capture) events are ignored until this bit is cleared. CAPSTS can be cleared by writing to CAPCLR bit in DCCAPCTL register and it re-triggers the capture mechanism. If active mode is enabled, via SHDWMODE bit in DCCAPCTL register, CPU reads of this register will return the active register value. If shadow mode is enabled, via SHDWMODE bit in DCCAPCTL register, the active register is copied to the shadow register on the PRD_eq or CNT_zero event (whichever is selected by PULSESEL bit in DCFCTL register). CPU reads of this register will return the shadow register value. Reset type: SYSRSn
14	CAPCLR	R-0/W1S	0h	DC Capture Latched Status Clear Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear this CAPSTS (set) condition. Reset type: SYSRSn
13	CAPSTS	R	0h	Latched Status Flag for Capture Event 0: No DC capture event occurred. 1: A DC capture event has occurred. Reset type: SYSRSn
12-2	RESERVED	R-0	0h	Reserved

Table 30-100. DCCAPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	SHDWMODE	R/W	0h	TBCTR Counter Capture Shadow Select Mode 0: Enable shadow mode. The DCCAP active register is copied to shadow register on a TBCTR = TBPRD or TBCTR = zero event as defined by the DCFCTL[PULSESEL] bit. CPU reads of the DCCAP register will return the shadow register contents. 1: Active Mode. In this mode the shadow register is disabled. CPU reads from the DCCAP register will always return the active register contents. Reset type: SYSRSn
0	CAPE	R/W	0h	TBCTR Counter Capture Enable/Disable 0: Disable the time-base counter capture. 1: Enable the time-base counter capture. Reset type: SYSRSn

30.20.2.75 DCFOFFSET Register (Offset = 192h) [Reset = 0000h]

DCFOFFSET is shown in Figure 30-196 and described in Table 30-101.

Return to the Summary Table.

Digital Compare Filter Offset Register

Figure 30-196. DCFOFFSET Register

15	14	13	12	11	10	9	8
DCFOFFSET							
R/W-0h							
7	6	5	4	3	2	1	0
DCFOFFSET							
R/W-0h							

Table 30-101. DCFOFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	DCFOFFSET	R/W	0h	Blanking Window Offset These 16-bits specify the number of TBCLK cycles from the blanking window reference to the point when the blanking window is applied. The blanking window reference is either period or zero as defined by the DCFCTL[PULSESEL] bit. This offset register is shadowed and the active register is loaded at the reference point defined by DCFCTL[PULSESEL]. The offset counter is also initialized and begins to count down when the active register is loaded. When the counter expires, the blanking window is applied. If the blanking window is currently active, then the blanking window counter is restarted. Reset type: SYSRSn

30.20.2.76 DCFOFFSETCNT Register (Offset = 194h) [Reset = 0000h]

DCFOFFSETCNT is shown in Figure 30-197 and described in Table 30-102.

Return to the Summary Table.

Digital Compare Filter Offset Counter Register

Figure 30-197. DCFOFFSETCNT Register

15	14	13	12	11	10	9	8
DCFOFFSETCNT							
R-0h							
7	6	5	4	3	2	1	0
DCFOFFSETCNT							
R-0h							

Table 30-102. DCFOFFSETCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	DCFOFFSETCNT	R	0h	Blanking Offset Counter These 16-bits are read only and indicate the current value of the offset counter. The counter counts down to zero and then stops until it is re-loaded on the next period or zero event as defined by the DCFCTL[PULSESEL] bit. The offset counter is not affected by the free/soft emulation bits. That is, it will always continue to count down if the device is halted by a emulation stop. Reset type: SYSRSn

30.20.2.77 DCFWINDOW Register (Offset = 196h) [Reset = 0000h]

DCFWINDOW is shown in Figure 30-198 and described in Table 30-103.

Return to the Summary Table.

Digital Compare Filter Window Register

Figure 30-198. DCFWINDOW Register

15	14	13	12	11	10	9	8
DCFWINDOW							
R/W-0h							
7	6	5	4	3	2	1	0
DCFWINDOW							
R/W-0h							

Table 30-103. DCFWINDOW Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	DCFWINDOW	R/W	0h	Blanking Window Width 00h: No blanking window is generated. 01-FFFFh: Specifies the width of the blanking window in TBCLK cycles. The blanking window begins when the offset counter expires. When this occurs, the window counter is loaded and begins to count down. If the blanking window is currently active and the offset counter expires, the blanking window counter is not restarted and the blanking window is cut short prematurely. Care should be taken to avoid this situation. The blanking window can cross a PWM period boundary. Reset type: SYSRSn

30.20.2.78 DCFWINDOWCNT Register (Offset = 198h) [Reset = 0000h]

DCFWINDOWCNT is shown in Figure 30-199 and described in Table 30-104.

Return to the Summary Table.

Digital Compare Filter Window Counter Register

Figure 30-199. DCFWINDOWCNT Register

15	14	13	12	11	10	9	8
DCFWINDOWCNT							
R-0h							
7	6	5	4	3	2	1	0
DCFWINDOWCNT							
R-0h							

Table 30-104. DCFWINDOWCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	DCFWINDOWCNT	R	0h	Blanking Window Counter These 16 bits are read only and indicate the current value of the window counter. The counter counts down to zero and then stops until it is re-loaded when the offset counter reaches zero again. Reset type: SYSRSn

30.20.2.79 BLANKPULSEMIXSEL Register (Offset = 19Ah) [Reset = 0000h]

BLANKPULSEMIXSEL is shown in Figure 30-200 and described in Table 30-105.

Return to the Summary Table.

Blanking window trigger pulse select register

Figure 30-200. BLANKPULSEMIXSEL Register

15	14	13	12	11	10	9	8
RESERVED						CDD	CDU
R-0-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CCD	CCU	CBD	CBU	CAD	CAU	PRD	ZRO
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-105. BLANKPULSEMIXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R-0	0h	Reserved
9	CDD	R/W	0h	Enable event time-base counter equal to CMPD when the timer is decrementing to the blanking window trigger (BLANKPULSEMIX). 0: CMPD down-count match enable event is not enabled 1: Enable CMPD down-count match enable event Reset type: SYSRSn
8	CDU	R/W	0h	Enable event time-base counter equal to CMPD when the timer is incrementing to the blanking window trigger (BLANKPULSEMIX). 0: CMPD up-count match enable event is not enabled 1: Enable CMPD up-count match enable event Reset type: SYSRSn
7	CCD	R/W	0h	Enable event time-base counter equal to CMPC when the timer is decrementing to the blanking window trigger (BLANKPULSEMIX). 0: CMPC down-count match enable event is not enabled 1: Enable CMPC down-count match enable event Reset type: SYSRSn
6	CCU	R/W	0h	Enable event time-base counter equal to CMPC when the timer is incrementing to the blanking window trigger (BLANKPULSEMIX). 0: CMPC up-count match enable event is not enabled 1: Enable CMPC up-count match enable event Reset type: SYSRSn
5	CBD	R/W	0h	Enable event time-base counter equal to CMPB when the timer is decrementing to the blanking window trigger (BLANKPULSEMIX). 0: CMPB down-count match enable event is not enabled 1: Enable CMPB down-count match enable event Reset type: SYSRSn
4	CBU	R/W	0h	Enable event time-base counter equal to CMPB when the timer is incrementing to the mixed ET interrupt trigger signal (BLANKPULSEMIX). 0: CMPB up-count match enable event is not enabled 1: Enable CMPB up-count match enable event Reset type: SYSRSn
3	CAD	R/W	0h	Enable event time-base counter equal to CMPA when the timer is decrementing to the blanking window trigger (BLANKPULSEMIX). 0: CMPA down-count match enable event is not enabled 1: Enable CMPA down-count match enable event Reset type: SYSRSn

Table 30-105. BLANKPULSEMIXSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	CAU	R/W	0h	Enable event time-base counter equal to CMPA when the timer is incrementing to the blanking window trigger (BLANKPULSEMIX). 0: CMPA up-count match enable event is not enabled 1: Enable CMPA up-count match enable event Reset type: SYSRSn
1	PRD	R/W	0h	Enable event time-base counter equal to period (TBCTR = TBPRD) to the blanking window trigger (BLANKPULSEMIX). 0: Period match event is not enabled 1: Enable period match event Reset type: SYSRSn
0	ZRO	R/W	0h	Enable event time-base counter equal to zero (TBCTR = 0x00) to the blanking window trigger (BLANKPULSEMIX). 0: Zero match event is not enabled 1: Enable zero match event Reset type: SYSRSn

30.20.2.80 DCCAPMIXSEL Register (Offset = 19Ch) [Reset = 0000h]

DCCAPMIXSEL is shown in Figure 30-201 and described in Table 30-106.

Return to the Summary Table.

Capture Event pulse select register

Figure 30-201. DCCAPMIXSEL Register

15	14	13	12	11	10	9	8
RESERVED						CDD	CDU
R-0-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CCD	CCU	CBD	CBU	CAD	CAU	PRD	ZRO
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-106. DCCAPMIXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R-0	0h	Reserved
9	CDD	R/W	0h	Enable event time-base counter equal to CMPD when the timer is decrementing to the blanking window trigger (DCCAPMIX). 0: CMPD down-count match enable event is not enabled 1: Enable CMPD down-count match enable event Reset type: SYSRSn
8	CDU	R/W	0h	Enable event time-base counter equal to CMPD when the timer is incrementing to the blanking window trigger (DCCAPMIX). 0: CMPD up-count match enable event is not enabled 1: Enable CMPD up-count match enable event Reset type: SYSRSn
7	CCD	R/W	0h	Enable event time-base counter equal to CMPC when the timer is decrementing to the blanking window trigger (DCCAPMIX). 0: CMPC down-count match enable event is not enabled 1: Enable CMPC down-count match enable event Reset type: SYSRSn
6	CCU	R/W	0h	Enable event time-base counter equal to CMPC when the timer is incrementing to the blanking window trigger (DCCAPMIX). 0: CMPC up-count match enable event is not enabled 1: Enable CMPC up-count match enable event Reset type: SYSRSn
5	CBD	R/W	0h	Enable event time-base counter equal to CMPB when the timer is decrementing to the blanking window trigger (DCCAPMIX). 0: CMPB down-count match enable event is not enabled 1: Enable CMPB down-count match enable event Reset type: SYSRSn
4	CBU	R/W	0h	Enable event time-base counter equal to CMPB when the timer is incrementing to the mixed ET interrupt trigger signal (DCCAPMIX). 0: CMPB up-count match enable event is not enabled 1: Enable CMPB up-count match enable event Reset type: SYSRSn
3	CAD	R/W	0h	Enable event time-base counter equal to CMPA when the timer is decrementing to the blanking window trigger (DCCAPMIX). 0: CMPA down-count match enable event is not enabled 1: Enable CMPA down-count match enable event Reset type: SYSRSn

Table 30-106. DCCAPMIXSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	CAU	R/W	0h	Enable event time-base counter equal to CMPA when the timer is incrementing to the blanking window trigger (DCCAPMIX). 0: CMPA up-count match enable event is not enabled 1: Enable CMPA up-count match enable event Reset type: SYSRSn
1	PRD	R/W	0h	Enable event time-base counter equal to period (TBCTR = TBPRD) to the blanking window trigger (DCCAPMIX). 0: Period match event is not enabled 1: Enable period match event Reset type: SYSRSn
0	ZRO	R/W	0h	Enable event time-base counter equal to zero (TBCTR = 0x00) to the blanking window trigger (DCCAPMIX). 0: Zero match event is not enabled 1: Enable zero match event Reset type: SYSRSn

30.20.2.81 DCCAP Register (Offset = 19Eh) [Reset = 0000h]

DCCAP is shown in Figure 30-202 and described in Table 30-107.

Return to the Summary Table.

Digital Compare Counter Capture Register

Figure 30-202. DCCAP Register

15	14	13	12	11	10	9	8
DCCAP							
R-0h							
7	6	5	4	3	2	1	0
DCCAP							
R-0h							

Table 30-107. DCCAP Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	DCCAP	R	0h	Digital Compare Time-Base Counter Capture To enable time-base counter capture, set the DCCAPCLT[CAPE] bit to 1. If enabled, reflects the value of the time-base counter (TBCTR) on the low to high edge transition of a filtered (DCEVTFLT) event. Further capture events are ignored until the next period or zero as selected by the DCFCTL[PULSESEL] bit. Shadowing of DCCAP is enabled and disabled by the DCCAPCTL[SHDWMODE] bit. By default this register is shadowed. - If DCCAPCTL[SHDWMODE] = 0, then the shadow is enabled. In this mode, the active register is copied to the shadow register on the TBCTR = TBPRD or TBCTR = zero as defined by the DCFCTL[PULSESEL] bit. CPU reads of this register will return the shadow register value. - If DCCAPCTL[SHDWMODE] = 1, then the shadow register is disabled. In this mode, CPU reads will return the active register value. The active and shadow registers share the same memory map address. Reset type: SYSRSn

30.20.2.82 DCAHTRIPSEL Register (Offset = 1A4h) [Reset = 0000h]

DCAHTRIPSEL is shown in Figure 30-203 and described in Table 30-108.

Return to the Summary Table.

Digital Compare AH Trip Select

Figure 30-203. DCAHTRIPSEL Register

15	14	13	12	11	10	9	8
RESERVED	TRIPINPUT15	TRIPINPUT14	RESERVED	TRIPINPUT12	TRIPINPUT11	TRIPINPUT10	TRIPINPUT9
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TRIPINPUT8	TRIPINPUT7	TRIPINPUT6	TRIPINPUT5	TRIPINPUT4	TRIPINPUT3	TRIPINPUT2	TRIPINPUT1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-108. DCAHTRIPSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	TRIPINPUT15	R/W	0h	TRIP Input 15 0: Trip Input 15 not selected as combinational ORed input 1: Trip Input 15 selected as combinational ORed input to DCAH mux Reset type: SYSRSn
13	TRIPINPUT14	R/W	0h	TRIP Input 14 0: Trip Input 14 not selected as combinational ORed input 1: Trip Input 14 selected as combinational ORed input to DCAH mux Reset type: SYSRSn
12	RESERVED	R/W	0h	Reserved
11	TRIPINPUT12	R/W	0h	TRIP Input 12 0: Trip Input 12 not selected as combinational ORed input 1: Trip Input 12 selected as combinational ORed input to DCAH mux Reset type: SYSRSn
10	TRIPINPUT11	R/W	0h	TRIP Input 11 0: Trip Input 11 not selected as combinational ORed input 1: Trip Input 11 selected as combinational ORed input to DCAH mux Reset type: SYSRSn
9	TRIPINPUT10	R/W	0h	TRIP Input 10 0: Trip Input 10 not selected as combinational ORed input 1: Trip Input 10 selected as combinational ORed input to DCAH mux Reset type: SYSRSn
8	TRIPINPUT9	R/W	0h	TRIP Input 9 0: Trip Input 9 not selected as combinational ORed input 1: Trip Input 9 selected as combinational ORed input to DCAH mux Reset type: SYSRSn
7	TRIPINPUT8	R/W	0h	TRIP Input 8 0: Trip Input 8 not selected as combinational ORed input 1: Trip Input 8 selected as combinational ORed input to DCAH mux Reset type: SYSRSn
6	TRIPINPUT7	R/W	0h	TRIP Input 7 0: Trip Input 7 not selected as combinational ORed input 1: Trip Input 7 selected as combinational ORed input to DCAH mux Reset type: SYSRSn
5	TRIPINPUT6	R/W	0h	TRIP Input 6 0: Trip Input 6 not selected as combinational ORed input 1: Trip Input 6 selected as combinational ORed input to DCAH mux Reset type: SYSRSn

Table 30-108. DCAHTRIPSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	TRIPINPUT5	R/W	0h	TRIP Input 5 0: Trip Input 5 not selected as combinational ORed input 1: Trip Input 5 selected as combinational ORed input to DCAH mux Reset type: SYSRSn
3	TRIPINPUT4	R/W	0h	TRIP Input 4 0: Trip Input 4 not selected as combinational ORed input 1: Trip Input 4 selected as combinational ORed input to DCAH mux Reset type: SYSRSn
2	TRIPINPUT3	R/W	0h	TRIP Input 3 0: Trip Input 3 not selected as combinational ORed input 1: Trip Input 3 selected as combinational ORed input to DCAH mux Reset type: SYSRSn
1	TRIPINPUT2	R/W	0h	TRIP Input 2 0: Trip Input 2 not selected as combinational ORed input 1: Trip Input 2 selected as combinational ORed input to DCAH mux Reset type: SYSRSn
0	TRIPINPUT1	R/W	0h	TRIP Input 1 0: Trip Input 1 not selected as combinational ORed input 1: Trip Input 1 selected as combinational ORed input to DCAH mux Reset type: SYSRSn

30.20.2.83 DCALTRIPSEL Register (Offset = 1A6h) [Reset = 0000h]

DCALTRIPSEL is shown in Figure 30-204 and described in Table 30-109.

Return to the Summary Table.

Digital Compare AL Trip Select

Figure 30-204. DCALTRIPSEL Register

15	14	13	12	11	10	9	8
RESERVED	TRIPINPUT15	TRIPINPUT14	RESERVED	TRIPINPUT12	TRIPINPUT11	TRIPINPUT10	TRIPINPUT9
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TRIPINPUT8	TRIPINPUT7	TRIPINPUT6	TRIPINPUT5	TRIPINPUT4	TRIPINPUT3	TRIPINPUT2	TRIPINPUT1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-109. DCALTRIPSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	TRIPINPUT15	R/W	0h	TRIP Input 15 0: Trip Input 15 not selected as combinational ORed input 1: Trip Input 15 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
13	TRIPINPUT14	R/W	0h	TRIP Input 14 0: Trip Input 14 not selected as combinational ORed input 1: Trip Input 14 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
12	RESERVED	R/W	0h	Reserved
11	TRIPINPUT12	R/W	0h	TRIP Input 12 0: Trip Input 12 not selected as combinational ORed input 1: Trip Input 12 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
10	TRIPINPUT11	R/W	0h	TRIP Input 11 0: Trip Input 11 not selected as combinational ORed input 1: Trip Input 11 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
9	TRIPINPUT10	R/W	0h	TRIP Input 10 0: Trip Input 10 not selected as combinational ORed input 1: Trip Input 10 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
8	TRIPINPUT9	R/W	0h	TRIP Input 9 0: Trip Input 9 not selected as combinational ORed input 1: Trip Input 9 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
7	TRIPINPUT8	R/W	0h	TRIP Input 8 0: Trip Input 8 not selected as combinational ORed input 1: Trip Input 8 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
6	TRIPINPUT7	R/W	0h	TRIP Input 7 0: Trip Input 7 not selected as combinational ORed input 1: Trip Input 7 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
5	TRIPINPUT6	R/W	0h	TRIP Input 6 0: Trip Input 6 not selected as combinational ORed input 1: Trip Input 6 selected as combinational ORed input to DCAL mux Reset type: SYSRSn

Table 30-109. DCALTRIPSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	TRIPINPUT5	R/W	0h	TRIP Input 5 0: Trip Input 5 not selected as combinational ORed input 1: Trip Input 5 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
3	TRIPINPUT4	R/W	0h	TRIP Input 4 0: Trip Input 4 not selected as combinational ORed input 1: Trip Input 4 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
2	TRIPINPUT3	R/W	0h	TRIP Input 3 0: Trip Input 3 not selected as combinational ORed input 1: Trip Input 3 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
1	TRIPINPUT2	R/W	0h	TRIP Input 2 0: Trip Input 2 not selected as combinational ORed input 1: Trip Input 2 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
0	TRIPINPUT1	R/W	0h	TRIP Input 1 0: Trip Input 1 not selected as combinational ORed input 1: Trip Input 1 selected as combinational ORed input to DCAL mux Reset type: SYSRSn

30.20.2.84 DCBHTRIPSEL Register (Offset = 1A8h) [Reset = 0000h]

DCBHTRIPSEL is shown in Figure 30-205 and described in Table 30-110.

Return to the Summary Table.

Digital Compare BH Trip Select

Figure 30-205. DCBHTRIPSEL Register

15	14	13	12	11	10	9	8
RESERVED	TRIPINPUT15	TRIPINPUT14	RESERVED	TRIPINPUT12	TRIPINPUT11	TRIPINPUT10	TRIPINPUT9
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TRIPINPUT8	TRIPINPUT7	TRIPINPUT6	TRIPINPUT5	TRIPINPUT4	TRIPINPUT3	TRIPINPUT2	TRIPINPUT1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-110. DCBHTRIPSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	TRIPINPUT15	R/W	0h	TRIP Input 15 0: Trip Input 15 not selected as combinational ORed input 1: Trip Input 15 selected as combinational ORed input to DCBH mux Reset type: SYSRSn
13	TRIPINPUT14	R/W	0h	TRIP Input 14 0: Trip Input 14 not selected as combinational ORed input 1: Trip Input 14 selected as combinational ORed input to DCBH mux Reset type: SYSRSn
12	RESERVED	R/W	0h	Reserved
11	TRIPINPUT12	R/W	0h	TRIP Input 12 0: Trip Input 12 not selected as combinational ORed input 1: Trip Input 12 selected as combinational ORed input to DCBH mux Reset type: SYSRSn
10	TRIPINPUT11	R/W	0h	TRIP Input 11 0: Trip Input 11 not selected as combinational ORed input 1: Trip Input 11 selected as combinational ORed input to DCBH mux Reset type: SYSRSn
9	TRIPINPUT10	R/W	0h	TRIP Input 10 0: Trip Input 10 not selected as combinational ORed input 1: Trip Input 10 selected as combinational ORed input to DCBH mux Reset type: SYSRSn
8	TRIPINPUT9	R/W	0h	TRIP Input 9 0: Trip Input 9 not selected as combinational ORed input 1: Trip Input 9 selected as combinational ORed input to DCBH mux Reset type: SYSRSn
7	TRIPINPUT8	R/W	0h	TRIP Input 8 0: Trip Input 8 not selected as combinational ORed input 1: Trip Input 8 selected as combinational ORed input to DCBH mux Reset type: SYSRSn
6	TRIPINPUT7	R/W	0h	TRIP Input 7 0: Trip Input 7 not selected as combinational ORed input 1: Trip Input 7 selected as combinational ORed input to DCBH mux Reset type: SYSRSn
5	TRIPINPUT6	R/W	0h	TRIP Input 6 0: Trip Input 6 not selected as combinational ORed input 1: Trip Input 6 selected as combinational ORed input to DCBH mux Reset type: SYSRSn

Table 30-110. DCBHTRIPSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	TRIPINPUT5	R/W	0h	TRIP Input 5 0: Trip Input 5 not selected as combinational ORed input 1: Trip Input 5 selected as combinational ORed input to DCBH mux Reset type: SYSRSn
3	TRIPINPUT4	R/W	0h	TRIP Input 4 0: Trip Input 4 not selected as combinational ORed input 1: Trip Input 4 selected as combinational ORed input to DCBH mux Reset type: SYSRSn
2	TRIPINPUT3	R/W	0h	TRIP Input 3 0: Trip Input 3 not selected as combinational ORed input 1: Trip Input 3 selected as combinational ORed input to DCBH mux Reset type: SYSRSn
1	TRIPINPUT2	R/W	0h	TRIP Input 2 0: Trip Input 2 not selected as combinational ORed input 1: Trip Input 2 selected as combinational ORed input to DCBH mux Reset type: SYSRSn
0	TRIPINPUT1	R/W	0h	TRIP Input 1 0: Trip Input 1 not selected as combinational ORed input 1: Trip Input 1 selected as combinational ORed input to DCBH mux Reset type: SYSRSn

30.20.2.85 DCBLTRIPSEL Register (Offset = 1AAh) [Reset = 0000h]

DCBLTRIPSEL is shown in Figure 30-206 and described in Table 30-111.

Return to the Summary Table.

Digital Compare BL Trip Select

Figure 30-206. DCBLTRIPSEL Register

15	14	13	12	11	10	9	8
RESERVED	TRIPINPUT15	TRIPINPUT14	RESERVED	TRIPINPUT12	TRIPINPUT11	TRIPINPUT10	TRIPINPUT9
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TRIPINPUT8	TRIPINPUT7	TRIPINPUT6	TRIPINPUT5	TRIPINPUT4	TRIPINPUT3	TRIPINPUT2	TRIPINPUT1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-111. DCBLTRIPSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	TRIPINPUT15	R/W	0h	TRIP Input 15 0: Trip Input 15 not selected as combinational ORed input 1: Trip Input 15 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
13	TRIPINPUT14	R/W	0h	TRIP Input 14 0: Trip Input 14 not selected as combinational ORed input 1: Trip Input 14 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
12	RESERVED	R/W	0h	Reserved
11	TRIPINPUT12	R/W	0h	TRIP Input 12 0: Trip Input 12 not selected as combinational ORed input 1: Trip Input 12 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
10	TRIPINPUT11	R/W	0h	TRIP Input 11 0: Trip Input 11 not selected as combinational ORed input 1: Trip Input 11 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
9	TRIPINPUT10	R/W	0h	TRIP Input 10 0: Trip Input 10 not selected as combinational ORed input 1: Trip Input 10 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
8	TRIPINPUT9	R/W	0h	TRIP Input 9 0: Trip Input 9 not selected as combinational ORed input 1: Trip Input 9 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
7	TRIPINPUT8	R/W	0h	TRIP Input 8 0: Trip Input 8 not selected as combinational ORed input 1: Trip Input 8 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
6	TRIPINPUT7	R/W	0h	TRIP Input 7 0: Trip Input 7 not selected as combinational ORed input 1: Trip Input 7 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
5	TRIPINPUT6	R/W	0h	TRIP Input 6 0: Trip Input 6 not selected as combinational ORed input 1: Trip Input 6 selected as combinational ORed input to DCAL mux Reset type: SYSRSn

Table 30-111. DCBLTRIPSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	TRIPINPUT5	R/W	0h	TRIP Input 5 0: Trip Input 5 not selected as combinational ORed input 1: Trip Input 5 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
3	TRIPINPUT4	R/W	0h	TRIP Input 4 0: Trip Input 4 not selected as combinational ORed input 1: Trip Input 4 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
2	TRIPINPUT3	R/W	0h	TRIP Input 3 0: Trip Input 3 not selected as combinational ORed input 1: Trip Input 3 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
1	TRIPINPUT2	R/W	0h	TRIP Input 2 0: Trip Input 2 not selected as combinational ORed input 1: Trip Input 2 selected as combinational ORed input to DCAL mux Reset type: SYSRSn
0	TRIPINPUT1	R/W	0h	TRIP Input 1 0: Trip Input 1 not selected as combinational ORed input 1: Trip Input 1 selected as combinational ORed input to DCAL mux Reset type: SYSRSn

30.20.2.86 CAPCTL Register (Offset = 1ACh) [Reset = 0000h]

CAPCTL is shown in Figure 30-207 and described in Table 30-112.

Return to the Summary Table.

Event Capture Control Register

Figure 30-207. CAPCTL Register

15	14	13	12	11	10	9	8
RESERVED							FRCLOAD
R-0h							R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED			PULSECTL	CAPINPOL	CAPGATEPOL		SRCSEL
R-0h			R/W-0h	R/W-0h	R/W-0h		R/W-0h

Table 30-112. CAPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8	FRCLOAD	R-0/W1S	0h	0: Writing of 0 is ignored. Always reads back a 0. 1: Forces a LOAD to occur on the DCCAP - an equivalent LOAD.active pulse Reset type: SYSRSn
7-5	RESERVED	R	0h	Reserved
4	PULSECTL	R/W	0h	Capture Input Polarity Select Mux: 0: Pulse selection determined by PULSESEL bits (common pulse selection for Blanking and Capture logic) 1: Pulse selection determined by CAPMIXSEL register (independent pulse selection for Blanking and Capture logic) Reset type: SYSRSn
3	CAPINPOL	R/W	0h	Capture Input Polarity Select Mux: 0: CAPIN.sync not inverted 1: CAPIN.sync Inverted Default state assumption for these inputs can be active high. If the user is providing active low signal then invert option can be configured Reset type: SYSRSn
2-1	CAPGATEPOL	R/W	0h	Capture Gate Input Polarity Select Mux: 00: Set to 1 - Gate is always ON 01: Set to 0 - Gate is always OFF 10: CAPGATE.sync 11: CAPGATE.sync Inverted Default state assumption for these inputs can be active high. If the user is providing active low signal then invert option can be configured Reset type: SYSRSn
0	SRCSEL	R/W	0h	Capture Logic Input Select Mux: 0: DCEVTFILT (Sync) - same as Type-4 1: CAPIN.sync Reset type: SYSRSn

30.20.2.87 CAPGATETRIPSEL Register (Offset = 1AEh) [Reset = 0000h]

CAPGATETRIPSEL is shown in Figure 30-208 and described in Table 30-113.

Return to the Summary Table.

Event Capture Gate Trip input select

Figure 30-208. CAPGATETRIPSEL Register

15	14	13	12	11	10	9	8
RESERVED	TRIPINPUT15	TRIPINPUT14	RESERVED	TRIPINPUT12	TRIPINPUT11	TRIPINPUT10	TRIPINPUT9
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TRIPINPUT8	TRIPINPUT7	TRIPINPUT6	TRIPINPUT5	TRIPINPUT4	TRIPINPUT3	TRIPINPUT2	TRIPINPUT1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-113. CAPGATETRIPSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	TRIPINPUT15	R/W	0h	TRIP Input 15 0: Trip Input 15 not selected as combinational ORed input 1: Trip Input 15 selected as combinational ORed input to CAPGATE mux Reset type: SYSRSn
13	TRIPINPUT14	R/W	0h	TRIP Input 14 0: Trip Input 14 not selected as combinational ORed input 1: Trip Input 14 selected as combinational ORed input to CAPGATE mux Reset type: SYSRSn
12	RESERVED	R/W	0h	Reserved
11	TRIPINPUT12	R/W	0h	TRIP Input 12 0: Trip Input 12 not selected as combinational ORed input 1: Trip Input 12 selected as combinational ORed input to CAPGATE mux Reset type: SYSRSn
10	TRIPINPUT11	R/W	0h	TRIP Input 11 0: Trip Input 11 not selected as combinational ORed input 1: Trip Input 11 selected as combinational ORed input to CAPGATE mux Reset type: SYSRSn
9	TRIPINPUT10	R/W	0h	TRIP Input 10 0: Trip Input 10 not selected as combinational ORed input 1: Trip Input 10 selected as combinational ORed input to CAPGATE mux Reset type: SYSRSn
8	TRIPINPUT9	R/W	0h	TRIP Input 9 0: Trip Input 9 not selected as combinational ORed input 1: Trip Input 9 selected as combinational ORed input to CAPGATE mux Reset type: SYSRSn
7	TRIPINPUT8	R/W	0h	TRIP Input 8 0: Trip Input 8 not selected as combinational ORed input 1: Trip Input 8 selected as combinational ORed input to CAPGATE mux Reset type: SYSRSn

Table 30-113. CAPGATETRIPSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	TRIPINPUT7	R/W	0h	TRIP Input 7 0: Trip Input 7 not selected as combinational ORed input 1: Trip Input 7 selected as combinational ORed input to CAPGATE mux Reset type: SYSRSn
5	TRIPINPUT6	R/W	0h	TRIP Input 6 0: Trip Input 6 not selected as combinational ORed input 1: Trip Input 6 selected as combinational ORed input to CAPGATE mux Reset type: SYSRSn
4	TRIPINPUT5	R/W	0h	TRIP Input 5 0: Trip Input 5 not selected as combinational ORed input 1: Trip Input 5 selected as combinational ORed input to CAPGATE mux Reset type: SYSRSn
3	TRIPINPUT4	R/W	0h	TRIP Input 4 0: Trip Input 4 not selected as combinational ORed input 1: Trip Input 4 selected as combinational ORed input to CAPGATE mux Reset type: SYSRSn
2	TRIPINPUT3	R/W	0h	TRIP Input 3 0: Trip Input 3 not selected as combinational ORed input 1: Trip Input 3 selected as combinational ORed input to CAPGATE mux Reset type: SYSRSn
1	TRIPINPUT2	R/W	0h	TRIP Input 2 0: Trip Input 2 not selected as combinational ORed input 1: Trip Input 2 selected as combinational ORed input to CAPGATE mux Reset type: SYSRSn
0	TRIPINPUT1	R/W	0h	TRIP Input 1 0: Trip Input 1 not selected as combinational ORed input 1: Trip Input 1 selected as combinational ORed input to CAPGATE mux Reset type: SYSRSn

30.20.2.88 CAPINTRIPSEL Register (Offset = 1B0h) [Reset = 0000h]

CAPINTRIPSEL is shown in Figure 30-209 and described in Table 30-114.

Return to the Summary Table.

Event Capture Trip input select

Figure 30-209. CAPINTRIPSEL Register

15	14	13	12	11	10	9	8
RESERVED	TRIPINPUT15	TRIPINPUT14	RESERVED	TRIPINPUT12	TRIPINPUT11	TRIPINPUT10	TRIPINPUT9
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TRIPINPUT8	TRIPINPUT7	TRIPINPUT6	TRIPINPUT5	TRIPINPUT4	TRIPINPUT3	TRIPINPUT2	TRIPINPUT1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-114. CAPINTRIPSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	TRIPINPUT15	R/W	0h	TRIP Input 15 0: Trip Input 15 not selected as combinational ORed input 1: Trip Input 15 selected as combinational ORed input to CAPIN mux Reset type: SYSRSn
13	TRIPINPUT14	R/W	0h	TRIP Input 14 0: Trip Input 14 not selected as combinational ORed input 1: Trip Input 14 selected as combinational ORed input to CAPIN mux Reset type: SYSRSn
12	RESERVED	R/W	0h	Reserved
11	TRIPINPUT12	R/W	0h	TRIP Input 12 0: Trip Input 12 not selected as combinational ORed input 1: Trip Input 12 selected as combinational ORed input to CAPIN mux Reset type: SYSRSn
10	TRIPINPUT11	R/W	0h	TRIP Input 11 0: Trip Input 11 not selected as combinational ORed input 1: Trip Input 11 selected as combinational ORed input to CAPIN mux Reset type: SYSRSn
9	TRIPINPUT10	R/W	0h	TRIP Input 10 0: Trip Input 10 not selected as combinational ORed input 1: Trip Input 10 selected as combinational ORed input to CAPIN mux Reset type: SYSRSn
8	TRIPINPUT9	R/W	0h	TRIP Input 9 0: Trip Input 9 not selected as combinational ORed input 1: Trip Input 9 selected as combinational ORed input to CAPIN mux Reset type: SYSRSn
7	TRIPINPUT8	R/W	0h	TRIP Input 8 0: Trip Input 8 not selected as combinational ORed input 1: Trip Input 8 selected as combinational ORed input to CAPIN mux Reset type: SYSRSn
6	TRIPINPUT7	R/W	0h	TRIP Input 7 0: Trip Input 7 not selected as combinational ORed input 1: Trip Input 7 selected as combinational ORed input to CAPIN mux Reset type: SYSRSn
5	TRIPINPUT6	R/W	0h	TRIP Input 6 0: Trip Input 6 not selected as combinational ORed input 1: Trip Input 6 selected as combinational ORed input to CAPIN mux Reset type: SYSRSn

Table 30-114. CAPINTRIPSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	TRIPINPUT5	R/W	0h	TRIP Input 5 0: Trip Input 5 not selected as combinational ORed input 1: Trip Input 5 selected as combinational ORed input to CAPIN mux Reset type: SYSRSn
3	TRIPINPUT4	R/W	0h	TRIP Input 4 0: Trip Input 4 not selected as combinational ORed input 1: Trip Input 4 selected as combinational ORed input to CAPIN mux Reset type: SYSRSn
2	TRIPINPUT3	R/W	0h	TRIP Input 3 0: Trip Input 3 not selected as combinational ORed input 1: Trip Input 3 selected as combinational ORed input to CAPIN mux Reset type: SYSRSn
1	TRIPINPUT2	R/W	0h	TRIP Input 2 0: Trip Input 2 not selected as combinational ORed input 1: Trip Input 2 selected as combinational ORed input to CAPIN mux Reset type: SYSRSn
0	TRIPINPUT1	R/W	0h	TRIP Input 1 0: Trip Input 1 not selected as combinational ORed input 1: Trip Input 1 selected as combinational ORed input to CAPIN mux Reset type: SYSRSn

30.20.2.89 CAPTRIPSEL Register (Offset = 1B2h) [Reset = 0000h]

CAPTRIPSEL is shown in Figure 30-210 and described in Table 30-115.

Return to the Summary Table.

Event Capture Signal Select

Figure 30-210. CAPTRIPSEL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CAPGATECOMPSEL				CAPINCOMPSEL			
R/W-0h				R/W-0h			

Table 30-115. CAPTRIPSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-4	CAPGATECOMPSEL	R/W	0h	Digital Compare A Low Input Select Bits 0000: TRIPIN1 0001: TRIPIN2 0010: TRIPIN3 0011: TRIPIN4 ... 1011: TRIPIN12 1100: Reserved 1101: TRIPIN14 1110: TRIPIN15 1111: Trip combination input (all trip inputs selected by CAPGATETRIPSEL register ORed together) Reset type: SYSRSn
3-0	CAPINCOMPSEL	R/W	0h	Digital Compare A High Input Select Bits 0000: TRIPIN1 0001: TRIPIN2 0010: TRIPIN3 0011: TRIPIN4 ... 1011: TRIPIN12 1100: Reserved 1101: TRIPIN14 1110: TRIPIN15 1111: Trip combination input (all trip inputs selected by CAPINTRIPSEL register ORed together) Reset type: SYSRSn

30.20.2.90 EPWMLOCK Register (Offset = 1F4h) [Reset = 00000000h]

EPWMLOCK is shown in Figure 30-211 and described in Table 30-116.

Return to the Summary Table.

EPWM Lock Register

Figure 30-211. EPWMLOCK Register

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
KEY							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			DCLOCK	TZCLRLOCK	TZCFGLOCK	GLLOCK	HRLOCK
R-0h			R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h

Table 30-116. EPWMLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write to this register succeeds only if this field is written with a value of 0xa5a5 Note: [1] Due to this KEY, only 32-bit writes will succeed (provided the KEY matches). 16-bit writes to the upper or lower half of this register will be ignored Reset type: SYSRSn
15-5	RESERVED	R	0h	Reserved
4	DCLOCK	R/WOnce	0h	0:Digital Compare registers from 0x180 to 0x1B2 offsets are protected by EALLOW. 1: Digital Compare registers from 0x180 and 0x1B2 offsets are locked and not writable. Reset type: SYSRSn
3	TZCLRLOCK	R/WOnce	0h	0:Trip Zone registers from 0x12E to 0x136 offsets are protected by EALLOW. 1: Trip Zone registers from 0x12E to 0x136 offsets are locked and not writable. Reset type: SYSRSn
2	TZCFGLOCK	R/WOnce	0h	0:TripZone registers from 0x100 to 0x11A and TZTRIPOUTSEL at 0x13A offsets are protected by EALLOW. 1: TripZone registers from 0x100 to 0x11A and TZTRIPOUTSEL at 0x13A offsets are locked and not writable. Reset type: SYSRSn
1	GLLOCK	R/WOnce	0h	0:Global Load registers from 0x68 to 0x6A offsets are protected by EALLOW. 1: Global Load registers from 0x68 to 0x6A offsets are locked and not writable Reset type: SYSRSn

Table 30-116. EPWMLOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	HRLOCK	R/WOnce	0h	0: HRPWM registers from 0x40 to 0x5A offsets are protected by EALLOW 1: HRPWM registers from 0x40 and 0x5A offsets are locked and not writable. Reset type: SYSRSn

30.20.2.91 HWVDELVAL Register (Offset = 1FAh) [Reset = 0000h]

HWVDELVAL is shown in Figure 30-212 and described in Table 30-117.

Return to the Summary Table.

Hardware Valley Mode Delay Register

Figure 30-212. HWVDELVAL Register

15	14	13	12	11	10	9	8
HWVDELVAL							
R-0h							
7	6	5	4	3	2	1	0
HWVDELVAL							
R-0h							

Table 30-117. HWVDELVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	HWVDELVAL	R	0h	Hardware Valley Delay Value Register This read only register reflects the hardware delay value calculated by the equations defined in VCAPCTL[VDELAYDIV]. This reflects the latest value from the hardware calculations and can change every time valley capture sequence is triggered and VCAP1 and VCAP2 values are updated. Reset type: SYSRSn

30.20.2.92 VCNTVAL Register (Offset = 1FCh) [Reset = 0000h]

VCNTVAL is shown in Figure 30-213 and described in Table 30-118.

Return to the Summary Table.

Hardware Valley Counter Register

Figure 30-213. VCNTVAL Register

15	14	13	12	11	10	9	8
VCNTVAL							
R-0h							
7	6	5	4	3	2	1	0
VCNTVAL							
R-0h							

Table 30-118. VCNTVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	VCNTVAL	R	0h	Valley Time Base Counter Register This register reflects the captured VCNT value upon occurrence of STOPEDGE selected in VCNTCFG register. Reset type: SYSRSn

30.20.3 EPWM_XCMP_REGS Registers

Table 30-119 lists the memory-mapped registers for the EPWM_XCMP_REGS registers. All register offset addresses not listed in Table 30-119 should be considered as reserved locations and the register contents should not be modified.

Table 30-119. EPWM_XCMP_REGS Registers

Offset	Acronym	Register Name	Protection
0h	XCMPCTL1	XCMP Mode Control Register	
10h	XLOADCTL	XCMP Mode Load Control Register	
18h	XLOAD	XCMP Mode Load Enable Register	
1Ch	EPWMXLINKXLOAD	Link register across PWM modules	
20h	XREGSHDW1STS	Shadow Buffer 1 Update Status Register	
28h	XREGSHDW2STS	Shadow Buffer 2 Update Status Register	
30h	XREGSHDW3STS	Shadow Buffer 3 Update Status Register	
200h	XCMP1_ACTIVE	Additional Compare 1 Active Register	
204h	XCMP2_ACTIVE	Additional Compare 2 Active Register	
208h	XCMP3_ACTIVE	Additional Compare 3 Active Register	
20Ch	XCMP4_ACTIVE	Additional Compare 4 Active Register	
210h	XCMP5_ACTIVE	Additional Compare 5 Active Register	
214h	XCMP6_ACTIVE	Additional Compare 6 Active Register	
218h	XCMP7_ACTIVE	Additional Compare 7 Active Register	
21Ch	XCMP8_ACTIVE	Additional Compare 8 Active Register	
220h	XTBPRD_ACTIVE	Additional Time Base Period Active Register	
230h	XAQCTLA_ACTIVE	AQCTLA Active Register	
232h	XAQCTLB_ACTIVE	AQCTLB Active Register	
244h	XMINMAX_ACTIVE	XMINMAX Active Register	
280h	XCMP1_SHDW1	Additional Compare 1 Shadow 1 Register	
284h	XCMP2_SHDW1	Additional Compare 2 Shadow 1 Register	
288h	XCMP3_SHDW1	Additional Compare 3 Shadow 1 Register	
28Ch	XCMP4_SHDW1	Additional Compare 4 Shadow 1 Register	
290h	XCMP5_SHDW1	Additional Compare 5 Shadow 1 Register	
294h	XCMP6_SHDW1	Additional Compare 6 Shadow 1 Register	
298h	XCMP7_SHDW1	Additional Compare 7 Shadow 1 Register	
29Ch	XCMP8_SHDW1	Additional Compare 8 Shadow 1 Register	
2A0h	XTBPRD_SHDW1	Additional Time Base Period Shadow 1 Register	
2B0h	XAQCTLA_SHDW1	XAQCTLA Shadow 1 Register	
2B2h	XAQCTLB_SHDW1	XAQCTLB Shadow 1 Register	
2BAh	CMPC_SHDW1	CMPC Shadow 1 Register	
2BEh	CMPD_SHDW1	CMPD Shadow 1 Register	
2C4h	XMINMAX_SHDW1	XMINMAX Shadow 1 Register	
300h	XCMP1_SHDW2	Additional Compare 1 Shadow 2 Register	
304h	XCMP2_SHDW2	Additional Compare 2 Shadow 2 Register	
308h	XCMP3_SHDW2	Additional Compare 3 Shadow 2 Register	
30Ch	XCMP4_SHDW2	Additional Compare 4 Shadow 2 Register	
310h	XCMP5_SHDW2	Additional Compare 5 Shadow 2 Register	
314h	XCMP6_SHDW2	Additional Compare 6 Shadow 2 Register	
318h	XCMP7_SHDW2	Additional Compare 7 Shadow 2 Register	

Table 30-119. EPWM_XCMP_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
31Ch	XCMP8_SHDW2	Additional Compare 8 Shadow 2 Register	
320h	XTBPRD_SHDW2	Additional Time Base Period Shadow 2 Register	
330h	XAQCTLA_SHDW2	XAQCTLA Shadow 2 Register	
332h	XAQCTLB_SHDW2	XAQCTLB Shadow 2 Register	
33Ah	CMPC_SHDW2	CMPC Shadow 2 Register	
33Eh	CMPD_SHDW2	CMPD Shadow 2 Register	
344h	XMINMAX_SHDW2	XMINMAX Shadow 2 Register	
380h	XCMP1_SHDW3	Additional Compare 1 Shadow 3 Register	
384h	XCMP2_SHDW3	Additional Compare 2 Shadow 3 Register	
388h	XCMP3_SHDW3	Additional Compare 3 Shadow 3 Register	
38Ch	XCMP4_SHDW3	Additional Compare 4 Shadow 3 Register	
390h	XCMP5_SHDW3	Additional Compare 5 Shadow 3 Register	
394h	XCMP6_SHDW3	Additional Compare 6 Shadow 3 Register	
398h	XCMP7_SHDW3	Additional Compare 7 Shadow 3 Register	
39Ch	XCMP8_SHDW3	Additional Compare 8 Shadow 3 Register	
3A0h	XTBPRD_SHDW3	Additional Time Base Period Shadow 3 Register	
3B0h	XAQCTLA_SHDW3	XAQCTLA Shadow 3 Register	
3B2h	XAQCTLB_SHDW3	XAQCTLB Shadow 3 Register	
3BAh	CMPC_SHDW3	CMPC Shadow 3 Register	
3BEh	CMPD_SHDW3	CMPD Shadow 3 Register	
3C4h	XMINMAX_SHDW3	XMINMAX Shadow 3 Register	

Complex bit access types are encoded to fit into small table cells. Table 30-120 shows the codes that are used for access types in this section.

Table 30-120. EPWM_XCMP_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

30.20.3.1 XCMPCTL1 Register (Offset = 0h) [Reset = 00000000h]

XCMPCTL1 is shown in Figure 30-214 and described in Table 30-121.

Return to the Summary Table.

XCMP Mode Control Register

Figure 30-214. XCMPCTL1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				XCMPB_ALLOC			
R-0-0h				R/W-0h			
7	6	5	4	3	2	1	0
XCMPA_ALLOC				RESERVED		XCMPSPLIT	XCMPEN
R/W-0h				R-0-0h		R/W-0h	R/W-0h

Table 30-121. XCMPCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-8	XCMPB_ALLOC	R/W	0h	XCMPn register allocation for CMPB: 0 : Reserved 1: Reserved 2 : Reserved 3: Reserved 4: Reserved 5: XCMP5 6: XCMP5, XCMP6 7: XCMP5, XCMP6, XCMP7 8: XCMP5, XCMP6, XCMP7, XCMP8 This register settings will take effect only when XCMPEN==1 And XCMPSPLIT ==1 Reset type: SYSRSn
7-4	XCMPA_ALLOC	R/W	0h	XCMPn register allocation for CMPA: 0: No XCMP 1: XCMP1 2: XCMP1, XCMP2 3: XCMP1, XCMP2, XCMP3 4: XCMP1, XCMP2, XCMP3, XCMP4 5: XCMP1, XCMP2, XCMP3, XCMP4, XCMP5 6: XCMP1, XCMP2, XCMP3, XCMP4, XCMP5, XCMP6 7: XCMP1, XCMP2, XCMP3, XCMP4, XCMP5, XCMP6, XCMP7 8: XCMP1, XCMP2, XCMP3, XCMP4, XCMP5, XCMP6, XCMP7, XCMP8 This register settings will take effect only when XCMPEN==1 If XCMPSPLIT ==1, this field cannot be greater than 4. If XCMPSPLIT ==1 only lower 3 bits are used in this field. Reset type: SYSRSn
3-2	RESERVED	R-0	0h	Reserved

Table 30-121. XCMPCTL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	XCMPSPLIT	R/W	0h	XCMP Register Allocation Options: 0 : XCMP1-8 --> CMPA 1 : XCMP1-4 -->CMPA, XCMP5-8 --> CMPB This register settings will take effect only when XCMPEN==1 Reset type: SYSRSn
0	XCMPEN	R/W	0h	XCMP Compare Register Operation Enable: 0: XCMP register operation Disabled (Operation compatible to Type-4) 1: XCMP register operation Enabled (New CMPx registers are effective) Reset type: SYSRSn

30.20.3.2 XLOADCTL Register (Offset = 10h) [Reset = 00000000h]

XLOADCTL is shown in Figure 30-215 and described in Table 30-122.

Return to the Summary Table.

XCMP Mode Load Control Register

Figure 30-215. XLOADCTL Register

31	30	29	28	27	26	25	24
RESERVED	RPTBUF3CNT			RESERVED	RPTBUF3PRD		
R-0-0h	R-0h			R-0-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	RPTBUF2CNT			RESERVED	RPTBUF2PRD		
R-0-0h	R-0h			R-0-0h	R/W-0h		
15	14	13	12	11	10	9	8
RESERVED				SHDWBUFPTR_LOADMULTIPLE		SHDWBUFPTR_LOADONCE	
R-0-0h				R-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED		SHDWLEVEL		RESERVED	LOADMODE	RESERVED	
R-0-0h		R/W-0h		R-0-0h	R/W-0h	R-0-0h	

Table 30-122. XLOADCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0	0h	Reserved
30-28	RPTBUF3CNT	R	0h	Repeat Count Status Shadow Buffer 3: These bits indicate how many times shadow buffer 3 has been be applied before moving to the next buffer I,e, shadow buffer 1. 000: Shadow buffer reset value with STARTLD and copied to Active register 001: Shadow buffer applied twice on 2 successive load strobes 010: Shadow buffer applied thrice on 3 successive load strobes . . 111: Shadow buffer applied 8 times on 8 successive load strobes These bits reset to zero every time STARTLD is initiated. Reset type: SYSRSn
27	RESERVED	R-0	0h	Reserved
26-24	RPTBUF3PRD	R/W	0h	Repeat Count Shadow Buffer 3 : These bits indicate how many times shadow buffer 3 will be applied before moving to the next buffer I,e, shadow buffer 1. 000: Apply shadow buffer once and move to the next shadow buffer on the following load pulse 001: Apply shadow buffer twice on 2 successive load strobes and move to the next shadow buffer on the following load pulse 010: Apply shadow buffer thrice on 3 successive load strobes and move to the next shadow buffer on the following load pulse . . 111: Apply shadow buffer 8 times on 8 successive load strobes and move to the next shadow buffer on the following load pulse Reset type: SYSRSn
23	RESERVED	R-0	0h	Reserved

Table 30-122. XLOADCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22-20	RPTBUF2CNT	R	0h	Repeat Count Status Shadow Buffer 2: These bits indicate how many times shadow buffer 2 has been be applied before moving to the next buffer I,e, shadow buffer 1. 000: Shadow buffer reset value with STARTLD and copied to Active register 001: Shadow buffer applied twice on 2 successive load strobes 010: Shadow buffer applied thrice on 3 successive load strobes . . 111: Shadow buffer applied 8 times on 8 successive load strobes These bits reset to zero every time STARTLD is initiated. Reset type: SYSRSn
19	RESERVED	R-0	0h	Reserved
18-16	RPTBUF2PRD	R/W	0h	Repeat Count Shadow Buffer 2 : These bits indicate how many times shadow buffer 2 will be applied before moving to the next buffer I,e, shadow buffer 1. 000: Apply shadow buffer once and move to the next shadow buffer on the following load pulse 001: Apply shadow buffer twice on 2 successive load strobes and move to the next shadow buffer on the following load pulse 010: Apply shadow buffer thrice on 3 successive load strobes and move to the next shadow buffer on the following load pulse . . 111: Apply shadow buffer 8 times on 8 successive load strobes and move to the next shadow buffer on the following load pulse Reset type: SYSRSn
15-12	RESERVED	R-0	0h	Reserved
11-10	SHDWBUFPTR_LOADMULTIPLE	R	0h	Register Load event count: These bits indicate the current shadow buffer in use. 00: Reset value 01: Shadow buffer 1 in use 10: Shadow buffer 2 in use 11: Shadow buffer 3 in use Reset type: SYSRSn
9-8	SHDWBUFPTR_LOADONCE	R/W	0h	Register Load event count: These bits indicate the current shadow buffer in use. 00: Reset value 01: Shadow buffer 1 in use 10: 2 Shadow buffer 2 in use 11: 3 Shadow buffer 3 in use Reset type: SYSRSn
7-6	RESERVED	R-0	0h	Reserved
5-4	SHDWLEVEL	R/W	0h	Shadow Register Level Allocation Options: These bits are effective only when XCMPEN is enabled. 00 : Shadow level is set at zero. Active register is available 01 : Shadow level is set at 1. SHDW1 and Active registers are available 10 : Shadow level is set at 1. SHDW1, SHDW2 and Active registers are available 11 : Shadow level is set at 1. SHDW1, SHDW2, SHDW3 and Active registers are available Reset type: SYSRSn
3	RESERVED	R-0	0h	Reserved

Table 30-122. XLOADCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	LOADMODE	R/W	0h	Load mode selection for Shadow registers: These bits are effective only when XCMPEN is enabled. 0 : (LOADONCE) Load occurs at every load strobe (CNT_Zero or FRCLD) from SHDWn Active registers. And STARTLD is cleared after 1 load strobe. SHDWBUFPTR is not automatically decremented in this case. SHDWBUFPTR needs to be set for subsequent loads. 1 : (LOADMULTIPLE) Load occurs at every load strobe (CNT_Zero or FRCLD) from SHDWn Active registers. And STARTLD is cleared after SHDWLEVEL number of load strobes. SHDWBUFPTR decrements by 1 on a load strobe, until the SHDWBUFPTR reaches 1. Reset type: SYSRSn
1-0	RESERVED	R-0	0h	Reserved

30.20.3.3 XLOAD Register (Offset = 18h) [Reset = 00000000h]

XLOAD is shown in Figure 30-216 and described in Table 30-123.

Return to the Summary Table.

XCMP Mode Load Enable Register

Figure 30-216. XLOAD Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						FRCLD	STARTLD
R-0-0h						R-0/W1S-0h	R-0/W1S-0h

Table 30-123. XLOAD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	FRCLD	R-0/W1S	0h	Force reload event in one shot mode : 1: Writing a 1 to this bit turn force one load event at the input of the event pre-scale counter as shown in the diagram below. This bit is intended to be used for testing and/or software force loading of the events in global load mode. 0: Writing of 0 will be ignored. Always reads back a 0. Reset type: SYSRSn
0	STARTLD	R-0/W1S	0h	Enable reload event : 1: Writing a 1 to this bit turn the one shot latch condition ON. Upon occurrence of a chosen load strobe, one shadow to active reload occurs and the latch will be cleared. Hence writing '1' to this bit would allow load strobe event to pass through and block further strobe events. 0: Writing of 0 will be ignored. Always reads back a 0. Reset type: SYSRSn

30.20.3.4 EPWMXLINKXLOAD Register (Offset = 1Ch) [Reset = 000000XXh]

EPWMXLINKXLOAD is shown in Figure 30-217 and described in Table 30-124.

Return to the Summary Table.

Link register across PWM modules

Figure 30-217. EPWMXLINKXLOAD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											XLOADLINK				
R-0-0h											R/W-Xh				

Table 30-124. EPWMXLINKXLOAD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R-0	0h	Reserved
4-0	XLOADLINK	R/W	Xh	XLOAD Link Bits: Writes to the XLOAD registers in the ePWM module selected by the following bit selections results in a simultaneous write to the current ePWM module's XLOAD registers 00000: ePWM1 00001: ePWM2 ... Up to the last instance of ePWM. All others are reserved. Reset type: SYSRSn

30.20.3.5 XREGSHDW1STS Register (Offset = 20h) [Reset = 00000000h]

XREGSHDW1STS is shown in Figure 30-218 and described in Table 30-125.

Return to the Summary Table.

Shadow Buffer 1 Update Status Register

Figure 30-218. XREGSHDW1STS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED	XMIN_SHDW1FULL	XMAX_SHDW1FULL	XAQCTLB_SHDW1FULL	XAQCTLA_SHDW1FULL	CMPD_SHDW1FULL	CMPC_SHDW1FULL	XTBPRD_SHDW1FULL
R-0-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
XCMP8_SHDW1FULL	XCMP7_SHDW1FULL	XCMP6_SHDW1FULL	XCMP5_SHDW1FULL	XCMP4_SHDW1FULL	XCMP3_SHDW1FULL	XCMP2_SHDW1FULL	XCMP1_SHDW1FULL
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 30-125. XREGSHDW1STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R-0	0h	Reserved
14	XMIN_SHDW1FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
13	XMAX_SHDW1FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
12	XAQCTLB_SHDW1FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
11	XAQCTLA_SHDW1FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
10	CMPD_SHDW1FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
9	CMPC_SHDW1FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
8	XTBPRD_SHDW1FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn

Table 30-125. XREGSHDW1STS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	XCMP8_SHDW1FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
6	XCMP7_SHDW1FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
5	XCMP6_SHDW1FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
4	XCMP5_SHDW1FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
3	XCMP4_SHDW1FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
2	XCMP3_SHDW1FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
1	XCMP2_SHDW1FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
0	XCMP1_SHDW1FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn

30.20.3.6 XREGSHDW2STS Register (Offset = 28h) [Reset = 00000000h]

XREGSHDW2STS is shown in Figure 30-219 and described in Table 30-126.

Return to the Summary Table.

Shadow Buffer 2 Update Status Register

Figure 30-219. XREGSHDW2STS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED	XMIN_SHDW2FULL	XMAX_SHDW2FULL	XAQCTLB_SHDW2FULL	XAQCTLA_SHDW2FULL	CMPD_SHDW2FULL	CMPC_SHDW2FULL	XTBPRD_SHDW2FULL
R-0-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
XCMP8_SHDW2FULL	XCMP7_SHDW2FULL	XCMP6_SHDW2FULL	XCMP5_SHDW2FULL	XCMP4_SHDW2FULL	XCMP3_SHDW2FULL	XCMP2_SHDW2FULL	XCMP1_SHDW2FULL
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 30-126. XREGSHDW2STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R-0	0h	Reserved
14	XMIN_SHDW2FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
13	XMAX_SHDW2FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
12	XAQCTLB_SHDW2FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
11	XAQCTLA_SHDW2FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
10	CMPD_SHDW2FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
9	CMPC_SHDW2FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
8	XTBPRD_SHDW2FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn

Table 30-126. XREGSHDW2STS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	XCMP8_SHDW2FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
6	XCMP7_SHDW2FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
5	XCMP6_SHDW2FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
4	XCMP5_SHDW2FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
3	XCMP4_SHDW2FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
2	XCMP3_SHDW2FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
1	XCMP2_SHDW2FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
0	XCMP1_SHDW2FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn

30.20.3.7 XREGSHDW3STS Register (Offset = 30h) [Reset = 00000000h]

XREGSHDW3STS is shown in Figure 30-220 and described in Table 30-127.

Return to the Summary Table.

Shadow Buffer 3 Update Status Register

Figure 30-220. XREGSHDW3STS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED	XMIN_SHDW3FULL	XMAX_SHDW3FULL	XAQCTLB_SHDW3FULL	XAQCTLA_SHDW3FULL	CMPD_SHDW3FULL	CMPC_SHDW3FULL	XTBPRD_SHDW3FULL
R-0-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
XCMP8_SHDW3FULL	XCMP7_SHDW3FULL	XCMP6_SHDW3FULL	XCMP5_SHDW3FULL	XCMP4_SHDW3FULL	XCMP3_SHDW3FULL	XCMP2_SHDW3FULL	XCMP1_SHDW3FULL
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 30-127. XREGSHDW3STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R-0	0h	Reserved
14	XMIN_SHDW3FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
13	XMAX_SHDW3FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
12	XAQCTLB_SHDW3FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
11	XAQCTLA_SHDW3FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
10	CMPD_SHDW3FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
9	CMPC_SHDW3FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
8	XTBPRD_SHDW3FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn

Table 30-127. XREGSHDW3STS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	XCMP8_SHDW3FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
6	XCMP7_SHDW3FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
5	XCMP6_SHDW3FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
4	XCMP5_SHDW3FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
3	XCMP4_SHDW3FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
2	XCMP3_SHDW3FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
1	XCMP2_SHDW3FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn
0	XCMP1_SHDW3FULL	R	0h	0 Shadow Register is not full yet 1 Indicates the Shadow Register is full, a CPU write will over-write current Shadow value Reset type: SYSRSn

30.20.3.8 XCMP1_ACTIVE Register (Offset = 200h) [Reset = 00000000h]

XCMP1_ACTIVE is shown in Figure 30-221 and described in Table 30-128.

Return to the Summary Table.

Additional Compare 1 Active Register

Figure 30-221. XCMP1_ACTIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP1_ACTIVE																XCMP1HR_ACTIVE															
R/W-0h																R/W-0h															

Table 30-128. XCMP1_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP1_ACTIVE	R/W	0h	XCMP1_ACTIVE Register The value in the XCMP1_ACTIVE register is loaded into CMPA/B (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP1HR_ACTIVE	R/W	0h	XCMP1HR_ACTIVE Register The value in the XCMP1HR_ACTIVE register is loaded into CMPA/BHR (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn

30.20.3.9 XCMP2_ACTIVE Register (Offset = 204h) [Reset = 00000000h]

XCMP2_ACTIVE is shown in Figure 30-222 and described in Table 30-129.

Return to the Summary Table.

Additional Compare 2 Active Register

Figure 30-222. XCMP2_ACTIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP2_ACTIVE																XCMP2HR_ACTIVE															
R/W-0h																R/W-0h															

Table 30-129. XCMP2_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP2_ACTIVE	R/W	0h	XCMP2_ACTIVE Register The value in the XCMP2_ACTIVE register is loaded into CMPA/B (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP2HR_ACTIVE	R/W	0h	XCMP2HR_ACTIVE Register The value in the XCMP2HR_ACTIVE register is loaded into CMPA/BHR (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn

30.20.3.10 XCMP3_ACTIVE Register (Offset = 208h) [Reset = 00000000h]

XCMP3_ACTIVE is shown in Figure 30-223 and described in Table 30-130.

Return to the Summary Table.

Additional Compare 3 Active Register

Figure 30-223. XCMP3_ACTIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP3_ACTIVE																XCMP3HR_ACTIVE															
R/W-0h																R/W-0h															

Table 30-130. XCMP3_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP3_ACTIVE	R/W	0h	XCMP3_ACTIVE Register The value in the XCMP3_ACTIVE register is loaded into CMPA/B (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP3HR_ACTIVE	R/W	0h	XCMP3HR_ACTIVE Register The value in the XCMP3HR_ACTIVE register is loaded into CMPA/BHR (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn

30.20.3.11 XCMP4_ACTIVE Register (Offset = 20Ch) [Reset = 00000000h]

XCMP4_ACTIVE is shown in Figure 30-224 and described in Table 30-131.

Return to the Summary Table.

Additional Compare 4 Active Register

Figure 30-224. XCMP4_ACTIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP4_ACTIVE																XCMP4HR_ACTIVE															
R/W-0h																R/W-0h															

Table 30-131. XCMP4_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP4_ACTIVE	R/W	0h	XCMP4_ACTIVE Register The value in the XCMP4_ACTIVE register is loaded into CMPA/B (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP4HR_ACTIVE	R/W	0h	XCMP4HR_ACTIVE Register The value in the XCMP4HR_ACTIVE register is loaded into CMPA/BHR (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn

30.20.3.12 XCMP5_ACTIVE Register (Offset = 210h) [Reset = 00000000h]

XCMP5_ACTIVE is shown in Figure 30-225 and described in Table 30-132.

Return to the Summary Table.

Additional Compare 5 Active Register

Figure 30-225. XCMP5_ACTIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP5_ACTIVE																XCMP5HR_ACTIVE															
R/W-0h																R/W-0h															

Table 30-132. XCMP5_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP5_ACTIVE	R/W	0h	XCMP5_ACTIVE Register The value in the XCMP5_ACTIVE register is loaded into CMPA/B (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP5HR_ACTIVE	R/W	0h	XCMP5HR_ACTIVE Register The value in the XCMP5HR_ACTIVE register is loaded into CMPA/BHR (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn

30.20.3.13 XCMP6_ACTIVE Register (Offset = 214h) [Reset = 00000000h]

XCMP6_ACTIVE is shown in Figure 30-226 and described in Table 30-133.

Return to the Summary Table.

Additional Compare 6 Active Register

Figure 30-226. XCMP6_ACTIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP6_ACTIVE																XCMP6HR_ACTIVE															
R/W-0h																R/W-0h															

Table 30-133. XCMP6_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP6_ACTIVE	R/W	0h	XCMP6_ACTIVE Register The value in the XCMP6_ACTIVE register is loaded into CMPA/B (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP6HR_ACTIVE	R/W	0h	XCMP6HR_ACTIVE Register The value in the XCMP6HR_ACTIVE register is loaded into CMPA/BHR (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn

30.20.3.14 XCMP7_ACTIVE Register (Offset = 218h) [Reset = 00000000h]

XCMP7_ACTIVE is shown in Figure 30-227 and described in Table 30-134.

Return to the Summary Table.

Additional Compare 7 Active Register

Figure 30-227. XCMP7_ACTIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP7_ACTIVE																XCMP7HR_ACTIVE															
R/W-0h																R/W-0h															

Table 30-134. XCMP7_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP7_ACTIVE	R/W	0h	XCMP7_ACTIVE Register The value in the XCMP7_ACTIVE register is loaded into CMPA/B (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP7HR_ACTIVE	R/W	0h	XCMP7HR_ACTIVE Register The value in the XCMP7HR_ACTIVE register is loaded into CMPA/BHR (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn

30.20.3.15 XCMP8_ACTIVE Register (Offset = 21Ch) [Reset = 00000000h]

XCMP8_ACTIVE is shown in Figure 30-228 and described in Table 30-135.

Return to the Summary Table.

Additional Compare 8 Active Register

Figure 30-228. XCMP8_ACTIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP8_ACTIVE																XCMP8HR_ACTIVE															
R/W-0h																R/W-0h															

Table 30-135. XCMP8_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP8_ACTIVE	R/W	0h	XCMP8_ACTIVE Register The value in the XCMP8_ACTIVE register is loaded into CMPA/B (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP8HR_ACTIVE	R/W	0h	XCMP8HR_ACTIVE Register The value in the XCMP8HR_ACTIVE register is loaded into CMPA/BHR (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn

30.20.3.16 XTBPRD_ACTIVE Register (Offset = 220h) [Reset = 00000000h]

XTBPRD_ACTIVE is shown in Figure 30-229 and described in Table 30-136.

Return to the Summary Table.

Additional Time Base Period Active Register

Figure 30-229. XTBPRD_ACTIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XTBPRD_ACTIVE																XTBPRDHR_ACTIVE															
R/W-0h																R/W-0h															

Table 30-136. XTBPRD_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XTBPRD_ACTIVE	R/W	0h	The value in the XTBPRD_ACTIVE register is loaded into TBPRD (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn
15-0	XTBPRDHR_ACTIVE	R/W	0h	The value in the XTBPRDHR_ACTIVE register is loaded into TBPRDHR (shadow/active) registers when shadow to active load occurs. Reset type: SYSRSn

30.20.3.17 XAQCTLA_ACTIVE Register (Offset = 230h) [Reset = 0000h]

XAQCTLA_ACTIVE is shown in Figure 30-230 and described in Table 30-137.

Return to the Summary Table.

AQCTLA Active Register

Figure 30-230. XAQCTLA_ACTIVE Register

15	14	13	12	11	10	9	8
XCMP8		XCMP7		XCMP6		XCMP5	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
XCMP4		XCMP3		XCMP2		XCMP1	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 30-137. XAQCTLA_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	XCMP8	R/W	0h	Action when Counter = CMP8 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
13-12	XCMP7	R/W	0h	Action when Counter = CMP7 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
11-10	XCMP6	R/W	0h	Action when Counter = CMP6 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
9-8	XCMP5	R/W	0h	Action when Counter = CMP5 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
7-6	XCMP4	R/W	0h	Action when Counter = CMP4 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
5-4	XCMP3	R/W	0h	Action when Counter = CMP3 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn

Table 30-137. XAQCTLA_ACTIVE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	XCMP2	R/W	0h	Action when Counter = CMP2 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
1-0	XCMP1	R/W	0h	Action when Counter = CMP1 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn

30.20.3.18 XAQCTLB_ACTIVE Register (Offset = 232h) [Reset = 0000h]

XAQCTLB_ACTIVE is shown in Figure 30-231 and described in Table 30-138.

Return to the Summary Table.

AQCTLB Active Register

Figure 30-231. XAQCTLB_ACTIVE Register

15	14	13	12	11	10	9	8
XCMP8		XCMP7		XCMP6		XCMP5	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED							
R-0-0h							

Table 30-138. XAQCTLB_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	XCMP8	R/W	0h	Action when Counter = CMP8 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
13-12	XCMP7	R/W	0h	Action when Counter = CMP7 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
11-10	XCMP6	R/W	0h	Action when Counter = CMP6 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
9-8	XCMP5	R/W	0h	Action when Counter = CMP5 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
7-0	RESERVED	R-0	0h	Reserved

30.20.3.19 XMINMAX_ACTIVE Register (Offset = 244h) [Reset = 00000000h]

XMINMAX_ACTIVE is shown in Figure 30-232 and described in Table 30-139.

Return to the Summary Table.

XMINMAX Active Register

Figure 30-232. XMINMAX_ACTIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XMIN_ACTIVE																XMAX_ACTIVE															
R/W-0h																R/W-0h															

Table 30-139. XMINMAX_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XMIN_ACTIVE	R/W	0h	The value in the XMIN_ACTIVE register is used for comparision against the threshold of the capture counter at any given time. Reset type: SYSRSn
15-0	XMAX_ACTIVE	R/W	0h	The value in the XMAX_ACTIVE register is used for comparision against the threshold of the capture counter at any given time. Reset type: SYSRSn

30.20.3.20 XCMP1_SHDW1 Register (Offset = 280h) [Reset = 00000000h]

XCMP1_SHDW1 is shown in Figure 30-233 and described in Table 30-140.

Return to the Summary Table.

Additional Compare 1 Shadow 1 Register

Figure 30-233. XCMP1_SHDW1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP1_SHDW1																XCMP1HR_SHDW1															
R/W-0h																R/W-0h															

Table 30-140. XCMP1_SHDW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP1_SHDW1	R/W	0h	XCMP1_SHDW1 Register The value in the XCMP1_SHDW1 register is loaded into XCMP1_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP1HR_SHDW1	R/W	0h	XCMP1HR_SHDW1 Register The value in the XCMP1HR_SHDW1 register is loaded into XCMP1HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.21 XCMP2_SHDW1 Register (Offset = 284h) [Reset = 00000000h]

XCMP2_SHDW1 is shown in Figure 30-234 and described in Table 30-141.

Return to the Summary Table.

Additional Compare 2 Shadow 1 Register

Figure 30-234. XCMP2_SHDW1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP2_SHDW1																XCMP2HR_SHDW1															
R/W-0h																R/W-0h															

Table 30-141. XCMP2_SHDW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP2_SHDW1	R/W	0h	XCMP2_SHDW1 Register The value in the XCMP2_SHDW1 register is loaded into XCMP2_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP2HR_SHDW1	R/W	0h	XCMP2HR_SHDW1 Register The value in the XCMP2HR_SHDW1 register is loaded into XCMP2HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.22 XCMP3_SHDW1 Register (Offset = 288h) [Reset = 00000000h]

XCMP3_SHDW1 is shown in Figure 30-235 and described in Table 30-142.

Return to the Summary Table.

Additional Compare 3 Shadow 1 Register

Figure 30-235. XCMP3_SHDW1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP3_SHDW1																XCMP3HR_SHDW1															
R/W-0h																R/W-0h															

Table 30-142. XCMP3_SHDW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP3_SHDW1	R/W	0h	XCMP3_SHDW1 Register The value in the XCMP3_SHDW1 register is loaded into XCMP3_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP3HR_SHDW1	R/W	0h	XCMP3HR_SHDW1 Register The value in the XCMP3HR_SHDW1 register is loaded into XCMP3HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.23 XCMP4_SHDW1 Register (Offset = 28Ch) [Reset = 00000000h]

XCMP4_SHDW1 is shown in Figure 30-236 and described in Table 30-143.

Return to the Summary Table.

Additional Compare 4 Shadow 1 Register

Figure 30-236. XCMP4_SHDW1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP4_SHDW1																XCMP4HR_SHDW1															
R/W-0h																R/W-0h															

Table 30-143. XCMP4_SHDW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP4_SHDW1	R/W	0h	XCMP4_SHDW1 Register The value in the XCMP4_SHDW1 register is loaded into XCMP4_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP4HR_SHDW1	R/W	0h	XCMP4HR_SHDW1 Register The value in the XCMP4HR_SHDW1 register is loaded into XCMP4HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.24 XCMP5_SHDW1 Register (Offset = 290h) [Reset = 00000000h]

XCMP5_SHDW1 is shown in Figure 30-237 and described in Table 30-144.

Return to the Summary Table.

Additional Compare 5 Shadow 1 Register

Figure 30-237. XCMP5_SHDW1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP5_SHDW1																XCMP5HR_SHDW1															
R/W-0h																R/W-0h															

Table 30-144. XCMP5_SHDW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP5_SHDW1	R/W	0h	XCMP5_SHDW1 Register The value in the XCMP5_SHDW1 register is loaded into XCMP5_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP5HR_SHDW1	R/W	0h	XCMP5HR_SHDW1 Register The value in the XCMP5HR_SHDW1 register is loaded into XCMP5HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.25 XCMP6_SHDW1 Register (Offset = 294h) [Reset = 00000000h]

XCMP6_SHDW1 is shown in Figure 30-238 and described in Table 30-145.

Return to the Summary Table.

Additional Compare 6 Shadow 1 Register

Figure 30-238. XCMP6_SHDW1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP6_SHDW1																XCMP6HR_SHDW1															
R/W-0h																R/W-0h															

Table 30-145. XCMP6_SHDW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP6_SHDW1	R/W	0h	XCMP6_SHDW1 Register The value in the XCMP6_SHDW1 register is loaded into XCMP6_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP6HR_SHDW1	R/W	0h	XCMP6HR_SHDW1 Register The value in the XCMP6HR_SHDW1 register is loaded into XCMP6HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.26 XCMP7_SHDW1 Register (Offset = 298h) [Reset = 00000000h]

XCMP7_SHDW1 is shown in Figure 30-239 and described in Table 30-146.

Return to the Summary Table.

Additional Compare 7 Shadow 1 Register

Figure 30-239. XCMP7_SHDW1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP7_SHDW1																XCMP7HR_SHDW1															
R/W-0h																R/W-0h															

Table 30-146. XCMP7_SHDW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP7_SHDW1	R/W	0h	XCMP7_SHDW1 Register The value in the XCMP7_SHDW1 register is loaded into XCMP7_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP7HR_SHDW1	R/W	0h	XCMP7HR_SHDW1 Register The value in the XCMP7HR_SHDW1 register is loaded into XCMP7HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.27 XCMP8_SHDW1 Register (Offset = 29Ch) [Reset = 00000000h]

XCMP8_SHDW1 is shown in Figure 30-240 and described in Table 30-147.

Return to the Summary Table.

Additional Compare 8 Shadow 1 Register

Figure 30-240. XCMP8_SHDW1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP8_SHDW1																XCMP8HR_SHDW1															
R/W-0h																R/W-0h															

Table 30-147. XCMP8_SHDW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP8_SHDW1	R/W	0h	XCMP8_SHDW1 Register The value in the XCMP8_SHDW1 register is loaded into XCMP8_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP8HR_SHDW1	R/W	0h	XCMP8HR_SHDW1 Register The value in the XCMP8HR_SHDW1 register is loaded into XCMP8HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.28 XTBPRD_SHDW1 Register (Offset = 2A0h) [Reset = 00000000h]

XTBPRD_SHDW1 is shown in Figure 30-241 and described in Table 30-148.

Return to the Summary Table.

Additional Time Base Period Shadow 1 Register

Figure 30-241. XTBPRD_SHDW1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XTBPRD_SHDW1																XTBPRDHR_SHDW1															
R/W-0h																R/W-0h															

Table 30-148. XTBPRD_SHDW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XTBPRD_SHDW1	R/W	0h	The value in the XTBPRD_SHDW1 register is loaded into XTBPRD_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XTBPRDHR_SHDW1	R/W	0h	The value in the XTBPRDHR_SHDW1 register is loaded into XTBPRDHR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.29 XAQCTLA_SHDW1 Register (Offset = 2B0h) [Reset = 0000h]

XAQCTLA_SHDW1 is shown in Figure 30-242 and described in Table 30-149.

Return to the Summary Table.

XAQCTLA Shadow 1 Register

Figure 30-242. XAQCTLA_SHDW1 Register

15	14	13	12	11	10	9	8
XCMP8		XCMP7		XCMP6		XCMP5	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
XCMP4		XCMP3		XCMP2		XCMP1	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 30-149. XAQCTLA_SHDW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	XCMP8	R/W	0h	Action when Counter = CMP8 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
13-12	XCMP7	R/W	0h	Action when Counter = CMP7 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
11-10	XCMP6	R/W	0h	Action when Counter = CMP6 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
9-8	XCMP5	R/W	0h	Action when Counter = CMP5 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
7-6	XCMP4	R/W	0h	Action when Counter = CMP4 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
5-4	XCMP3	R/W	0h	Action when Counter = CMP3 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn

Table 30-149. XAQCTLA_SHDW1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	XCMP2	R/W	0h	Action when Counter = CMP2 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
1-0	XCMP1	R/W	0h	Action when Counter = CMP1 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn

30.20.3.30 XAQCTLB_SHDW1 Register (Offset = 2B2h) [Reset = 0000h]

XAQCTLB_SHDW1 is shown in Figure 30-243 and described in Table 30-150.

Return to the Summary Table.

XAQCTLB Shadow 1 Register

Figure 30-243. XAQCTLB_SHDW1 Register

15	14	13	12	11	10	9	8
XCMP8		XCMP7		XCMP6		XCMP5	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED							
R-0-0h							

Table 30-150. XAQCTLB_SHDW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	XCMP8	R/W	0h	Action when Counter = CMP8 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
13-12	XCMP7	R/W	0h	Action when Counter = CMP7 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
11-10	XCMP6	R/W	0h	Action when Counter = CMP6 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
9-8	XCMP5	R/W	0h	Action when Counter = CMP5 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
7-0	RESERVED	R-0	0h	Reserved

30.20.3.31 CMPC_SHDW1 Register (Offset = 2BAh) [Reset = 0000h]

CMPC_SHDW1 is shown in Figure 30-244 and described in Table 30-151.

Return to the Summary Table.

CMPC Shadow 1 Register

Figure 30-244. CMPC_SHDW1 Register

15	14	13	12	11	10	9	8
CMPC_SHDW1							
R/W-0h							
7	6	5	4	3	2	1	0
CMPC_SHDW1							
R/W-0h							

Table 30-151. CMPC_SHDW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CMPC_SHDW1	R/W	0h	The value in the CMPC_SHDW1 register is loaded into CMPC_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.32 CMPD_SHDW1 Register (Offset = 2BEh) [Reset = 0000h]

CMPD_SHDW1 is shown in Figure 30-245 and described in Table 30-152.

Return to the Summary Table.

CMPD Shadow 1 Register

Figure 30-245. CMPD_SHDW1 Register

15	14	13	12	11	10	9	8
CMPD_SHDW1							
R/W-0h							
7	6	5	4	3	2	1	0
CMPD_SHDW1							
R/W-0h							

Table 30-152. CMPD_SHDW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CMPD_SHDW1	R/W	0h	The value in the CMPD_SHDW1 register is loaded into CMPD_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.33 XMINMAX_SHDW1 Register (Offset = 2C4h) [Reset = 00000000h]

XMINMAX_SHDW1 is shown in Figure 30-246 and described in Table 30-153.

Return to the Summary Table.

XMINMAX Shadow 1 Register

Figure 30-246. XMINMAX_SHDW1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XMIN_SHDW1																XMAX_SHDW1															
R/W-0h																R/W-0h															

Table 30-153. XMINMAX_SHDW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XMIN_SHDW1	R/W	0h	The value in the XMIN_SHDW1 register is loaded into XMIN_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XMAX_SHDW1	R/W	0h	The value in the XMAX_SHDW1 register is loaded into XMAX_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.34 XCMP1_SHDW2 Register (Offset = 300h) [Reset = 00000000h]

XCMP1_SHDW2 is shown in Figure 30-247 and described in Table 30-154.

Return to the Summary Table.

Additional Compare 1 Shadow 2 Register

Figure 30-247. XCMP1_SHDW2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP1_SHDW2																XCMP1HR_SHDW2															
R/W-0h																R/W-0h															

Table 30-154. XCMP1_SHDW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP1_SHDW2	R/W	0h	XCMP1_SHDW2 Register The value in the XCMP1_SHDW2 register is loaded into XCMP1_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP1HR_SHDW2	R/W	0h	XCMP1HR_SHDW2 Register The value in the XCMP1HR_SHDW2 register is loaded into XCMP1HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.35 XCMP2_SHDW2 Register (Offset = 304h) [Reset = 00000000h]

XCMP2_SHDW2 is shown in Figure 30-248 and described in Table 30-155.

Return to the Summary Table.

Additional Compare 2 Shadow 2 Register

Figure 30-248. XCMP2_SHDW2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP2_SHDW2																XCMP2HR_SHDW2															
R/W-0h																R/W-0h															

Table 30-155. XCMP2_SHDW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP2_SHDW2	R/W	0h	XCMP2_SHDW2 Register The value in the XCMP2_SHDW2 register is loaded into XCMP2_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP2HR_SHDW2	R/W	0h	XCMP2HR_SHDW2 Register The value in the XCMP2HR_SHDW2 register is loaded into XCMP2HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.36 XCMP3_SHDW2 Register (Offset = 308h) [Reset = 00000000h]

XCMP3_SHDW2 is shown in Figure 30-249 and described in Table 30-156.

Return to the Summary Table.

Additional Compare 3 Shadow 2 Register

Figure 30-249. XCMP3_SHDW2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP3_SHDW2																XCMP3HR_SHDW2															
R/W-0h																R/W-0h															

Table 30-156. XCMP3_SHDW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP3_SHDW2	R/W	0h	XCMP3_SHDW2 Register The value in the XCMP3_SHDW2 register is loaded into XCMP3_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP3HR_SHDW2	R/W	0h	XCMP3HR_SHDW2 Register The value in the XCMP3HR_SHDW2 register is loaded into XCMP3HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.37 XCMP4_SHDW2 Register (Offset = 30Ch) [Reset = 00000000h]

XCMP4_SHDW2 is shown in Figure 30-250 and described in Table 30-157.

Return to the Summary Table.

Additional Compare 4 Shadow 2 Register

Figure 30-250. XCMP4_SHDW2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP4_SHDW2																XCMP4HR_SHDW2															
R/W-0h																R/W-0h															

Table 30-157. XCMP4_SHDW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP4_SHDW2	R/W	0h	XCMP4_SHDW2 Register The value in the XCMP4_SHDW2 register is loaded into XCMP4_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP4HR_SHDW2	R/W	0h	XCMP4HR_SHDW2 Register The value in the XCMP4HR_SHDW2 register is loaded into XCMP4HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.38 XCMP5_SHDW2 Register (Offset = 310h) [Reset = 00000000h]

XCMP5_SHDW2 is shown in Figure 30-251 and described in Table 30-158.

Return to the Summary Table.

Additional Compare 5 Shadow 2 Register

Figure 30-251. XCMP5_SHDW2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP5_SHDW2																XCMP5HR_SHDW2															
R/W-0h																R/W-0h															

Table 30-158. XCMP5_SHDW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP5_SHDW2	R/W	0h	XCMP5_SHDW2 Register The value in the XCMP5_SHDW2 register is loaded into XCMP5_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP5HR_SHDW2	R/W	0h	XCMP5HR_SHDW2 Register The value in the XCMP5HR_SHDW2 register is loaded into XCMP5HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.39 XCMP6_SHDW2 Register (Offset = 314h) [Reset = 00000000h]

XCMP6_SHDW2 is shown in Figure 30-252 and described in Table 30-159.

Return to the Summary Table.

Additional Compare 6 Shadow 2 Register

Figure 30-252. XCMP6_SHDW2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP6_SHDW2																XCMP6HR_SHDW2															
R/W-0h																R/W-0h															

Table 30-159. XCMP6_SHDW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP6_SHDW2	R/W	0h	XCMP6_SHDW2 Register The value in the XCMP6_SHDW2 register is loaded into XCMP6_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP6HR_SHDW2	R/W	0h	XCMP6HR_SHDW2 Register The value in the XCMP6HR_SHDW2 register is loaded into XCMP6HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.40 XCMP7_SHDW2 Register (Offset = 318h) [Reset = 00000000h]

XCMP7_SHDW2 is shown in Figure 30-253 and described in Table 30-160.

Return to the Summary Table.

Additional Compare 7 Shadow 2 Register

Figure 30-253. XCMP7_SHDW2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP7_SHDW2																XCMP7HR_SHDW2															
R/W-0h																R/W-0h															

Table 30-160. XCMP7_SHDW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP7_SHDW2	R/W	0h	XCMP7_SHDW2 Register The value in the XCMP7_SHDW2 register is loaded into XCMP7_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP7HR_SHDW2	R/W	0h	XCMP7HR_SHDW2 Register The value in the XCMP7HR_SHDW2 register is loaded into XCMP7HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.41 XCMP8_SHDW2 Register (Offset = 31Ch) [Reset = 00000000h]

XCMP8_SHDW2 is shown in Figure 30-254 and described in Table 30-161.

Return to the Summary Table.

Additional Compare 8 Shadow 2 Register

Figure 30-254. XCMP8_SHDW2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP8_SHDW2																XCMP8HR_SHDW2															
R/W-0h																R/W-0h															

Table 30-161. XCMP8_SHDW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP8_SHDW2	R/W	0h	XCMP8_SHDW2 Register The value in the XCMP8_SHDW2 register is loaded into XCMP8_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP8HR_SHDW2	R/W	0h	XCMP8HR_SHDW2 Register The value in the XCMP8HR_SHDW2 register is loaded into XCMP8HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.42 XTBPRD_SHDW2 Register (Offset = 320h) [Reset = 00000000h]

XTBPRD_SHDW2 is shown in Figure 30-255 and described in Table 30-162.

Return to the Summary Table.

Additional Time Base Period Shadow 2 Register

Figure 30-255. XTBPRD_SHDW2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XTBPRD_SHDW2																XTBPRDHR_SHDW2															
R/W-0h																R/W-0h															

Table 30-162. XTBPRD_SHDW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XTBPRD_SHDW2	R/W	0h	The value in the XTBPRD_SHDW2 register is loaded into XTBPRD_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XTBPRDHR_SHDW2	R/W	0h	The value in the XTBPRDHR_SHDW2 register is loaded into XTBPRDHR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.43 XAQCTLA_SHDW2 Register (Offset = 330h) [Reset = 0000h]

XAQCTLA_SHDW2 is shown in Figure 30-256 and described in Table 30-163.

Return to the Summary Table.

XAQCTLA Shadow 2 Register

Figure 30-256. XAQCTLA_SHDW2 Register

15	14	13	12	11	10	9	8
XCMP8		XCMP7		XCMP6		XCMP5	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
XCMP4		XCMP3		XCMP2		XCMP1	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 30-163. XAQCTLA_SHDW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	XCMP8	R/W	0h	Action when Counter = CMP8 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
13-12	XCMP7	R/W	0h	Action when Counter = CMP7 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
11-10	XCMP6	R/W	0h	Action when Counter = CMP6 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
9-8	XCMP5	R/W	0h	Action when Counter = CMP5 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
7-6	XCMP4	R/W	0h	Action when Counter = CMP4 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
5-4	XCMP3	R/W	0h	Action when Counter = CMP3 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn

Table 30-163. XAQCTLA_SHDW2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	XCMP2	R/W	0h	Action when Counter = CMP2 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
1-0	XCMP1	R/W	0h	Action when Counter = CMP1 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn

30.20.3.44 XAQCTLB_SHDW2 Register (Offset = 332h) [Reset = 0000h]

XAQCTLB_SHDW2 is shown in Figure 30-257 and described in Table 30-164.

Return to the Summary Table.

XAQCTLB Shadow 2 Register

Figure 30-257. XAQCTLB_SHDW2 Register

15	14	13	12	11	10	9	8
XCMP8		XCMP7		XCMP6		XCMP5	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED							
R-0-0h							

Table 30-164. XAQCTLB_SHDW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	XCMP8	R/W	0h	Action when Counter = CMP8 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
13-12	XCMP7	R/W	0h	Action when Counter = CMP7 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
11-10	XCMP6	R/W	0h	Action when Counter = CMP6 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
9-8	XCMP5	R/W	0h	Action when Counter = CMP5 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
7-0	RESERVED	R-0	0h	Reserved

30.20.3.45 CMPC_SHDW2 Register (Offset = 33Ah) [Reset = 0000h]

CMPC_SHDW2 is shown in Figure 30-258 and described in Table 30-165.

Return to the Summary Table.

CMPC Shadow 2 Register

Figure 30-258. CMPC_SHDW2 Register

15	14	13	12	11	10	9	8
CMPC_SHDW2							
R/W-0h							
7	6	5	4	3	2	1	0
CMPC_SHDW2							
R/W-0h							

Table 30-165. CMPC_SHDW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CMPC_SHDW2	R/W	0h	The value in the CMPC_SHDW2 register is loaded into CMPC_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.46 CMPD_SHDW2 Register (Offset = 33Eh) [Reset = 0000h]

CMPD_SHDW2 is shown in Figure 30-259 and described in Table 30-166.

Return to the Summary Table.

CMPD Shadow 2 Register

Figure 30-259. CMPD_SHDW2 Register

15	14	13	12	11	10	9	8
CMPD_SHDW2							
R/W-0h							
7	6	5	4	3	2	1	0
CMPD_SHDW2							
R/W-0h							

Table 30-166. CMPD_SHDW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CMPD_SHDW2	R/W	0h	The value in the CMPD_SHDW2 register is loaded into CMPD_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.47 XMINMAX_SHDW2 Register (Offset = 344h) [Reset = 00000000h]

XMINMAX_SHDW2 is shown in Figure 30-260 and described in Table 30-167.

Return to the Summary Table.

XMINMAX Shadow 2 Register

Figure 30-260. XMINMAX_SHDW2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XMIN_SHDW2																XMAX_SHDW2															
R/W-0h																R/W-0h															

Table 30-167. XMINMAX_SHDW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XMIN_SHDW2	R/W	0h	The value in the XMIN_SHDW2 register is loaded into XMIN_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XMAX_SHDW2	R/W	0h	The value in the XMAX_SHDW2 register is loaded into XMAX_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.48 XCMP1_SHDW3 Register (Offset = 380h) [Reset = 00000000h]

XCMP1_SHDW3 is shown in Figure 30-261 and described in Table 30-168.

Return to the Summary Table.

Additional Compare 1 Shadow 3 Register

Figure 30-261. XCMP1_SHDW3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP1_SHDW3																XCMP1HR_SHDW3															
R/W-0h																R/W-0h															

Table 30-168. XCMP1_SHDW3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP1_SHDW3	R/W	0h	XCMP1_SHDW3 Register The value in the XCMP1_SHDW3 register is loaded into XCMP1_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP1HR_SHDW3	R/W	0h	XCMP1HR_SHDW3 Register The value in the XCMP1HR_SHDW3 register is loaded into XCMP1HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.49 XCMP2_SHDW3 Register (Offset = 384h) [Reset = 00000000h]

XCMP2_SHDW3 is shown in Figure 30-262 and described in Table 30-169.

Return to the Summary Table.

Additional Compare 2 Shadow 3 Register

Figure 30-262. XCMP2_SHDW3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP2_SHDW3																XCMP2HR_SHDW3															
R/W-0h																R/W-0h															

Table 30-169. XCMP2_SHDW3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP2_SHDW3	R/W	0h	XCMP2_SHDW3 Register The value in the XCMP2_SHDW3 register is loaded into XCMP2_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP2HR_SHDW3	R/W	0h	XCMP2HR_SHDW3 Register The value in the XCMP2HR_SHDW3 register is loaded into XCMP2HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.50 XCMP3_SHDW3 Register (Offset = 388h) [Reset = 00000000h]

XCMP3_SHDW3 is shown in Figure 30-263 and described in Table 30-170.

Return to the Summary Table.

Additional Compare 3 Shadow 3 Register

Figure 30-263. XCMP3_SHDW3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP3_SHDW3																XCMP3HR_SHDW3															
R/W-0h																R/W-0h															

Table 30-170. XCMP3_SHDW3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP3_SHDW3	R/W	0h	XCMP3_SHDW3 Register The value in the XCMP3_SHDW3 register is loaded into XCMP3_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP3HR_SHDW3	R/W	0h	XCMP3HR_SHDW3 Register The value in the XCMP3HR_SHDW3 register is loaded into XCMP3HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.51 XCMP4_SHDW3 Register (Offset = 38Ch) [Reset = 00000000h]

XCMP4_SHDW3 is shown in Figure 30-264 and described in Table 30-171.

Return to the Summary Table.

Additional Compare 4 Shadow 3 Register

Figure 30-264. XCMP4_SHDW3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP4_SHDW3																XCMP4HR_SHDW3															
R/W-0h																R/W-0h															

Table 30-171. XCMP4_SHDW3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP4_SHDW3	R/W	0h	XCMP4_SHDW3 Register The value in the XCMP4_SHDW3 register is loaded into XCMP4_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP4HR_SHDW3	R/W	0h	XCMP4HR_SHDW3 Register The value in the XCMP4HR_SHDW3 register is loaded into XCMP4HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.52 XCMP5_SHDW3 Register (Offset = 390h) [Reset = 00000000h]

XCMP5_SHDW3 is shown in Figure 30-265 and described in Table 30-172.

Return to the Summary Table.

Additional Compare 5 Shadow 3 Register

Figure 30-265. XCMP5_SHDW3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP5_SHDW3																XCMP5HR_SHDW3															
R/W-0h																R/W-0h															

Table 30-172. XCMP5_SHDW3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP5_SHDW3	R/W	0h	XCMP5_SHDW3 Register The value in the XCMP5_SHDW3 register is loaded into XCMP5_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP5HR_SHDW3	R/W	0h	XCMP5HR_SHDW3 Register The value in the XCMP5HR_SHDW3 register is loaded into XCMP5HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.53 XCMP6_SHDW3 Register (Offset = 394h) [Reset = 00000000h]

XCMP6_SHDW3 is shown in Figure 30-266 and described in Table 30-173.

Return to the Summary Table.

Additional Compare 6 Shadow 3 Register

Figure 30-266. XCMP6_SHDW3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP6_SHDW3																XCMP6HR_SHDW3															
R/W-0h																R/W-0h															

Table 30-173. XCMP6_SHDW3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP6_SHDW3	R/W	0h	XCMP6_SHDW3 Register The value in the XCMP6_SHDW3 register is loaded into XCMP6_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP6HR_SHDW3	R/W	0h	XCMP6HR_SHDW3 Register The value in the XCMP6HR_SHDW3 register is loaded into XCMP6HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.54 XCMP7_SHDW3 Register (Offset = 398h) [Reset = 00000000h]

XCMP7_SHDW3 is shown in Figure 30-267 and described in Table 30-174.

Return to the Summary Table.

Additional Compare 7 Shadow 3 Register

Figure 30-267. XCMP7_SHDW3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP7_SHDW3																XCMP7HR_SHDW3															
R/W-0h																R/W-0h															

Table 30-174. XCMP7_SHDW3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP7_SHDW3	R/W	0h	XCMP7_SHDW3 Register The value in the XCMP7_SHDW3 register is loaded into XCMP7_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP7HR_SHDW3	R/W	0h	XCMP7HR_SHDW3 Register The value in the XCMP7HR_SHDW3 register is loaded into XCMP7HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.55 XCMP8_SHDW3 Register (Offset = 39Ch) [Reset = 00000000h]

XCMP8_SHDW3 is shown in Figure 30-268 and described in Table 30-175.

Return to the Summary Table.

Additional Compare 8 Shadow 3 Register

Figure 30-268. XCMP8_SHDW3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCMP8_SHDW3																XCMP8HR_SHDW3															
R/W-0h																R/W-0h															

Table 30-175. XCMP8_SHDW3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XCMP8_SHDW3	R/W	0h	XCMP8_SHDW3 Register The value in the XCMP8_SHDW3 register is loaded into XCMP8_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XCMP8HR_SHDW3	R/W	0h	XCMP8HR_SHDW3 Register The value in the XCMP8HR_SHDW3 register is loaded into XCMP8HR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.56 XTBPRD_SHDW3 Register (Offset = 3A0h) [Reset = 00000000h]

XTBPRD_SHDW3 is shown in Figure 30-269 and described in Table 30-176.

Return to the Summary Table.

Additional Time Base Period Shadow 3 Register

Figure 30-269. XTBPRD_SHDW3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XTBPRD_SHDW3																XTBPRDHR_SHDW3															
R/W-0h																R/W-0h															

Table 30-176. XTBPRD_SHDW3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XTBPRD_SHDW3	R/W	0h	The value in the XTBPRD_SHDW3 register is loaded into XTBPRD_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XTBPRDHR_SHDW3	R/W	0h	The value in the XTBPRDHR_SHDW3 register is loaded into XTBPRDHR_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.57 XAQCTLA_SHDW3 Register (Offset = 3B0h) [Reset = 0000h]

XAQCTLA_SHDW3 is shown in Figure 30-270 and described in Table 30-177.

Return to the Summary Table.

XAQCTLA Shadow 3 Register

Figure 30-270. XAQCTLA_SHDW3 Register

15	14	13	12	11	10	9	8
XCMP8		XCMP7		XCMP6		XCMP5	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
XCMP4		XCMP3		XCMP2		XCMP1	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 30-177. XAQCTLA_SHDW3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	XCMP8	R/W	0h	Action when Counter = CMP8 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
13-12	XCMP7	R/W	0h	Action when Counter = CMP7 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
11-10	XCMP6	R/W	0h	Action when Counter = CMP6 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
9-8	XCMP5	R/W	0h	Action when Counter = CMP5 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
7-6	XCMP4	R/W	0h	Action when Counter = CMP4 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
5-4	XCMP3	R/W	0h	Action when Counter = CMP3 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn

Table 30-177. XAQCTLA_SHDW3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	XCMP2	R/W	0h	Action when Counter = CMP2 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
1-0	XCMP1	R/W	0h	Action when Counter = CMP1 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn

30.20.3.58 XAQCTLB_SHDW3 Register (Offset = 3B2h) [Reset = 0000h]

XAQCTLB_SHDW3 is shown in Figure 30-271 and described in Table 30-178.

Return to the Summary Table.

XAQCTLB Shadow 3 Register

Figure 30-271. XAQCTLB_SHDW3 Register

15	14	13	12	11	10	9	8
XCMP8		XCMP7		XCMP6		XCMP5	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED							
R-0-0h							

Table 30-178. XAQCTLB_SHDW3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	XCMP8	R/W	0h	Action when Counter = CMP8 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
13-12	XCMP7	R/W	0h	Action when Counter = CMP7 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
11-10	XCMP6	R/W	0h	Action when Counter = CMP6 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
9-8	XCMP5	R/W	0h	Action when Counter = CMP5 00: Do nothing (action disabled) 01: Clear (low) 10: Set (high) 11: Toggle (Low -> High, High -> Low) Reset type: SYSRSn
7-0	RESERVED	R-0	0h	Reserved

30.20.3.59 CMPC_SHDW3 Register (Offset = 3BAh) [Reset = 0000h]

CMPC_SHDW3 is shown in Figure 30-272 and described in Table 30-179.

Return to the Summary Table.

CMPC Shadow 3 Register

Figure 30-272. CMPC_SHDW3 Register

15	14	13	12	11	10	9	8
CMPC_SHDW3							
R/W-0h							
7	6	5	4	3	2	1	0
CMPC_SHDW3							
R/W-0h							

Table 30-179. CMPC_SHDW3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CMPC_SHDW3	R/W	0h	The value in the CMPC_SHDW3 register is loaded into CMPC_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.60 CMPD_SHDW3 Register (Offset = 3BEh) [Reset = 0000h]

CMPD_SHDW3 is shown in Figure 30-273 and described in Table 30-180.

Return to the Summary Table.

CMPD Shadow 3 Register

Figure 30-273. CMPD_SHDW3 Register

15	14	13	12	11	10	9	8
CMPD_SHDW3							
R/W-0h							
7	6	5	4	3	2	1	0
CMPD_SHDW3							
R/W-0h							

Table 30-180. CMPD_SHDW3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CMPD_SHDW3	R/W	0h	The value in the CMPD_SHDW3 register is loaded into CMPD_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.3.61 XMINMAX_SHDW3 Register (Offset = 3C4h) [Reset = 00000000h]

XMINMAX_SHDW3 is shown in Figure 30-274 and described in Table 30-181.

Return to the Summary Table.

XMINMAX Shadow 3 Register

Figure 30-274. XMINMAX_SHDW3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XMIN_SHDW3																XMAX_SHDW3															
R/W-0h																R/W-0h															

Table 30-181. XMINMAX_SHDW3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	XMIN_SHDW3	R/W	0h	The value in the XMIN_SHDW3 register is loaded into XMIN_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn
15-0	XMAX_SHDW3	R/W	0h	The value in the XMAX_SHDW3 register is loaded into XMAX_ACTIVE register when shadow to active load occurs. Reset type: SYSRSn

30.20.4 DE_REGS Registers

Table 30-182 lists the memory-mapped registers for the DE_REGS registers. All register offset addresses not listed in Table 30-182 should be considered as reserved locations and the register contents should not be modified.

Table 30-182. DE_REGS Registers

Offset	Acronym	Register Name	Protection
0h	DECTL	DE control register	
4h	DECOMPSEL	DE comparator source select register	
8h	DEACTCTL	DE Action Control	
Ch	DESTS	DE Status register	
10h	DEFRC	DE Status force register	
14h	DECLR	DE Status clear register	
20h	DEMONCNT	DE trip monitor counter	
24h	DEMONCTL	DE monitor mode control	
28h	DEMONSTEP	DE monitor counter step	
2Ch	DEMONTHRES	DE monitor counter threshold	

Complex bit access types are encoded to fit into small table cells. Table 30-183 shows the codes that are used for access types in this section.

Table 30-183. DE_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

30.20.4.1 DECTL Register (Offset = 0h) [Reset = 00000000h]

DECTL is shown in Figure 30-275 and described in Table 30-184.

Return to the Summary Table.

DE control register

Figure 30-275. DECTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
REENTRYDLY							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						MODE	ENABLE
R-0h						R/W-0h	R/W-0h

Table 30-184. DECTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	REENTRYDLY	R/W	0h	Determines the blocking window after DEACTIVE flag is cleared in which setting of DEACTIVE flag is prevented from being set. 0 : No blocking 1 : Blocked until 1 EPWMSYNCPER event 2 : Blocked until 2 EPWMSYNCPER events . . 255 : Blocked until 255 EPWMSYNCPER events Reset type: SYSRSn
7-2	RESERVED	R	0h	Reserved
1	MODE	R/W	0h	0 : DEACTIVE flag works in cycle by cycle mode. On every EPWMSYNCPER, set condition of DEACTIVE flag is evaluated. If the set condition is not present the flag is cleared. 1 : DEACTIVE flag works in one shot mode (hardware set) and software clear. Reset type: SYSRSn
0	ENABLE	R/W	0h	DE function enable 0 : Diode Emulation mode functionality is disabled. DEACTIVE flag is not set on a TRIPH_OR_TRIPL event. 1 : Diode Emulation mode functionality is enabled. DEACTIVE flag is set on a TRIPH_OR_TRIPL event. Note: ENABLE bit is cleared on a EPWMTRIPOUT event. Software has to re-enable this bit after EPWMTRIPOUT condition is serviced. Reset type: SYSRSn

30.20.4.2 DECOMPSEL Register (Offset = 4h) [Reset = 00000000h]

DECOMPSEL is shown in Figure 30-276 and described in Table 30-185.

Return to the Summary Table.

Used to configure the comparator whose trip sources will be used.

Figure 30-276. DECOMPSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										TRIPH						RESERVED										TRIPL					
R-0h										R/W-0h						R-0h										R/W-0h					

Table 30-185. DECOMPSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	Reserved
21-16	TRIPH	R/W	0h	These bits determine the source of the TRIPH signal. Other Values defined in the 'ePWM DE TripH Selection' table Note: All the reserved encodings result in TRIPH being 0. Reset type: SYSRSn
15-6	RESERVED	R	0h	Reserved
5-0	TRIPL	R/W	0h	These bits determine the source of the TRIPL signal. Other Values defined in the 'ePWM DE TripL Selection' table Note: All the reserved encodings result in TRIPL being 0. Reset type: SYSRSn

30.20.4.3 DEACTCTL Register (Offset = 8h) [Reset = 00000000h]

DEACTCTL is shown in Figure 30-277 and described in Table 30-186.

Return to the Summary Table.

Used to configure the PWM controls when in DE mode.

Figure 30-277. DEACTCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							TRIPENABLE
R-0h							R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	TRIPSELB	PWMB		RESERVED	TRIPSELA	PWMA	
R-0h	R/W-0h	R/W-0h		R-0h	R/W-0h	R/W-0h	

Table 30-186. DEACTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	Reserved
16	TRIPENABLE	R/W	0h	0 : EPWMTRIPOUT does not bypass the diode emulation logic. 1 : EPWMTRIPOUT bypasses the diode emulation ePWM generation logic (not complete bypass of module) Reset type: SYSRSn
15-7	RESERVED	R	0h	Reserved
6	TRIPSELB	R/W	0h	0 : TRIPH 1 : TRIPL Reset type: SYSRSn
5-4	PWMB	R/W	0h	00 : synchronized version of TRIPH or TRIPL signal as selected by the TRIPSELB 01 : synchronized and inverted version of TRIPH or TRIPL signal as selected by the TRIPSELB 10 : A constant 0 drives PWMB when DEACTIVE flag is set. 11 : A constant 1 drives PWMB when DEACTIVE flag is set. Reset type: SYSRSn
3	RESERVED	R	0h	Reserved
2	TRIPSELA	R/W	0h	0 : TRIPH 1 : TRIPL Reset type: SYSRSn
1-0	PWMA	R/W	0h	00 : synchronized version of TRIPH or TRIPL signal as selected by the TRIPSELA 01 : synchronized and inverted version of TRIPH or TRIPL signal as selected by the TRIPSELA 10 : A constant 0 drives PWMA when DEACTIVE flag is set. 11 : A constant 1 drives PWMA when DEACTIVE flag is set. Reset type: SYSRSn

30.20.4.4 DESTS Register (Offset = Ch) [Reset = 00000000h]

DESTS is shown in Figure 30-278 and described in Table 30-187.

Return to the Summary Table.

DE Status register

Figure 30-278. DESTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							DEACTIVE
R-0h							R-0h

Table 30-187. DESTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	DEACTIVE	R	0h	0 : Diode emulation mode is not active 1 : Diode emulation mode is active Reset type: SYSRSn

30.20.4.5 DEFRC Register (Offset = 10h) [Reset = 00000000h]

DEFRC is shown in Figure 30-279 and described in Table 30-188.

Return to the Summary Table.

DE Status force register

Figure 30-279. DEFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							DEACTIVE
R-0h							R-0/W1S-0h

Table 30-188. DEFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	DEACTIVE	R-0/W1S	0h	0 : No effect. 1 : Forces DEACTIVE flag to 1. Reset type: SYSRSn

30.20.4.6 DECLR Register (Offset = 14h) [Reset = 00000000h]

DECLR is shown in Figure 30-280 and described in Table 30-189.

Return to the Summary Table.

DE Status clear register

Figure 30-280. DECLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							DEACTIVE
R-0h							R-0/W1S-0h

Table 30-189. DECLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	DEACTIVE	R-0/W1S	0h	0 : No effect. 1 : Clears DEACTIVE flag. Reset type: SYSRSn

30.20.4.7 DEMONCNT Register (Offset = 20h) [Reset = 00000000h]

DEMONCNT is shown in Figure 30-281 and described in Table 30-190.

Return to the Summary Table.

DE trip monitor counter

Figure 30-281. DEMONCNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CNT															
R-0h																R-0h															

Table 30-190. DEMONCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved

15-0	CNT	R	0h	An 16-bit counter which monitors the frequency of diode mode trip events.
				When TripHorTripL is active: Increment CNT (increment INCSTEP on every EPWMxSYNC)
				When TripHorTripL is in-active: Decrement CNT (decrement DECSTEP on every EPWMxSYNC)
				If(CNT > THRESHOLD) then generate DETRIP and clear the counter.
				If((CNT - DECSTEP) < 0) then CNT = 0
				If((CNT + INCSTEP) >= 0xFFFF) then CNT = 0xFFFF
				Note : CNT is cleared when DECTL.ENABLE is 0
				Note: DEMONTHRES == 0x0 should not generate trip as the DEMONTHRES and DEMONCNT registers have reset value of 0x0
				Reset type: SYSRSn

30.20.4.8 DEMONCTL Register (Offset = 24h) [Reset = 00000000h]

DEMONCTL is shown in Figure 30-282 and described in Table 30-191.

Return to the Summary Table.

DE monitor mode control

Figure 30-282. DEMONCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							ENABLE
R-0h							R/W-0h

Table 30-191. DEMONCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	ENABLE	R/W	0h	Enable bit for DE Mode Monitor counter function. 0: DE Mode Monitor counter function is disabled 1: DE Mode Monitor counter function is enabled Reset type: SYSRSn

30.20.4.9 DEMONSTEP Register (Offset = 28h) [Reset = 00000000h]

DEMONSTEP is shown in Figure 30-283 and described in Table 30-192.

Return to the Summary Table.

DE monitor counter step

Figure 30-283. DEMONSTEP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DECSTEP								RESERVED								INCSTEP							
R-0h								R/W-0h								R-0h								R/W-0h							

Table 30-192. DEMONSTEP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	DECSTEP	R/W	0h	Defines the decrement step of DEMONCNT[CNT] counter. Reset type: SYSRSn
15-8	RESERVED	R	0h	Reserved
7-0	INCSTEP	R/W	0h	Defines the increment step of DEMONCNT[CNT] counter. Reset type: SYSRSn

30.20.4.10 DEMONTHRES Register (Offset = 2Ch) [Reset = 00000000h]

DEMONTHRES is shown in Figure 30-284 and described in Table 30-193.

Return to the Summary Table.

DE monitor counter threshold

Figure 30-284. DEMONTHRES Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																THRESHOLD															
R-0h																R/W-0h															

Table 30-193. DEMONTHRES Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	THRESHOLD	R/W	0h	Defines the threshold of DE monitor counter. Reset type: SYSRSn

30.20.5 MINDB_LUT_REGS Registers

Table 30-194 lists the memory-mapped registers for the MINDB_LUT_REGS registers. All register offset addresses not listed in Table 30-194 should be considered as reserved locations and the register contents should not be modified.

Table 30-194. MINDB_LUT_REGS Registers

Offset	Acronym	Register Name	Protection
0h	MINDBCFG	Minimum dead band configuration register.	
4h	MINDBDLY	Minimum dead band delay register	
20h	LUTCTLA	LUT control register on PWMA	
24h	LUTCTLB	LUT control register on PWMB	

Complex bit access types are encoded to fit into small table cells. Table 30-195 shows the codes that are used for access types in this section.

Table 30-195. MINDB_LUT_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

30.20.5.1 MINDBCFG Register (Offset = 0h) [Reset = 00000000h]

MINDBCFG is shown in Figure 30-285 and described in Table 30-196.

Return to the Summary Table.

Minimum dead band configuration register.

Figure 30-285. MINDBCFG Register

31	30	29	28	27	26	25	24
RESERVED							POLSELB
R-0h							R/W-0h
23	22	21	20	19	18	17	16
SELB				SELBLOCKB	INVERTB	RESERVED	ENABLEB
R/W-0h				R/W-0h	R/W-0h	R-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							POLSELA
R-0h							R/W-0h
7	6	5	4	3	2	1	0
SELA				SELBLOCKA	INVERTA	RESERVED	ENABLEA
R/W-0h				R/W-0h	R/W-0h	R-0h	R/W-0h

Table 30-196. MINDBCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	POLSELB	R/W	0h	Select signal for the AND OR logic of BLOCKB (output of SELBLOCKB mux) and PWMB signals 0 : Select BLOCKB is inverted and ANDed with PWMB. 1 : Select BLOCKB is Ored with PWMB. Reset type: SYSRSn
23-20	SELB	R/W	0h	PWMB min dead band reference 0x0 : EPWMxB_DB_NO_HR 0x1 : Output 1 from MINDB XBAR 0x2 : Output 2 from MINDB XBAR . . 0xf : Output 15 from MINDB XBAR Reset type: SYSRSn
19	SELBLOCKB	R/W	0h	0 : Select BLOCKB as the blocking signal on PWMB. 1 : Select BLOCKA as the blocking signal on PWMB. Reset type: SYSRSn
18	INVERTB	R/W	0h	0 : No inversion on the selected reference signal which is used in the min deadband logic on PWMB. 1 : Invert the selected reference signal which is used in the min deadband logic on PWMB. Reset type: SYSRSn
17	RESERVED	R	0h	Reserved
16	ENABLEB	R/W	0h	0 : Minimum dead band logic is disabled 1 : Minimum dead band logic is enabled Reset type: SYSRSn
15-9	RESERVED	R	0h	Reserved

Table 30-196. MINDBCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	POLSELA	R/W	0h	Select signal for the AND OR logic of BLOCKA (output of SELBLOCKA mux) and PWMA signals 0 : Select BLOCKA is inverted and ANDed with PWMA. 1 : Select BLOCKA is Ored with PWMA. Reset type: SYSRSn
7-4	SELA	R/W	0h	PWMA min dead band reference 0x0 : EPWMxA_DB_NO_HR 0x1 : Output 1 from MINDB XBAR 0x2 : Output 2 from MINDB XBAR . . 0xf : Output 15 from MINDB XBAR Reset type: SYSRSn
3	SELBLOCKA	R/W	0h	0 : Select BLOCKA as the blocking signal on PWMA. 1 : Select BLOCKB as the blocking signal on PWMB. Reset type: SYSRSn
2	INVERTA	R/W	0h	0 : No inversion on the selected reference signal which is used in the min deadband logic on PWMA. 1 : Invert the selected reference signal which is used in the min deadband logic on PWMA. Reset type: SYSRSn
1	RESERVED	R	0h	Reserved
0	ENABLEA	R/W	0h	0 : Minimum dead band logic is disabled 1 : Minimum dead band logic is enabled Reset type: SYSRSn

30.20.5.2 MINDBDLY Register (Offset = 4h) [Reset = 00000000h]

MINDBDLY is shown in Figure 30-286 and described in Table 30-197.

Return to the Summary Table.

Minimum dead band delay register

Figure 30-286. MINDBDLY Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DELAYB																DELAYA															
R/W-0h																R/W-0h															

Table 30-197. MINDBDLY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DELAYB	R/W	0h	Minimum dead band delay on PWMB in terms of SYSCLK cycles. For delay value of 0, MINDBCFG[ENABLEA/B] = '0' should be configured. If MINDBCFG[ENABLEA/B] = '1' and MINDBDLY[DELAYA/B]='0' then delay is '1' cycle is applied. Reset type: SYSRSn
15-0	DELAYA	R/W	0h	Minimum dead band delay on PWMA in terms of SYSCLK cycles. For delay value of 0, MINDBCFG[ENABLEA/B] = '0' should be configured. If MINDBCFG[ENABLEA/B] = '1' and MINDBDLY[DELAYA/B]='0' then delay is '1' cycle is applied. Reset type: SYSRSn

30.20.5.3 LUTCTLA Register (Offset = 20h) [Reset = 00000001h]

LUTCTLA is shown in Figure 30-287 and described in Table 30-198.

Return to the Summary Table.

LUT control register on PWMA

Figure 30-287. LUTCTLA Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
LUTDEC7	LUTDEC6	LUTDEC5	LUTDEC4	LUTDEC3	LUTDEC2	LUTDEC1	LUTDEC0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
SELXBAR				RESERVED			BYPASS
R/W-0h				R-0h			R/W-1h

Table 30-198. LUTCTLA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23	LUTDEC7	R/W	0h	0 : Force 0 1 : Force 1 Reset type: SYSRSn
22	LUTDEC6	R/W	0h	0 : Force 0 1 : Force 1 Reset type: SYSRSn
21	LUTDEC5	R/W	0h	0 : Force 0 1 : Force 1 Reset type: SYSRSn
20	LUTDEC4	R/W	0h	0 : Force 0 1 : Force 1 Reset type: SYSRSn
19	LUTDEC3	R/W	0h	0 : Force 0 1 : Force 1 Reset type: SYSRSn
18	LUTDEC2	R/W	0h	0 : Force 0 1 : Force 1 Reset type: SYSRSn
17	LUTDEC1	R/W	0h	0 : Force 0 1 : Force 1 Reset type: SYSRSn
16	LUTDEC0	R/W	0h	0 : Force 0 1 : Force 1 Reset type: SYSRSn
15-8	RESERVED	R	0h	Reserved
7-4	SELXBAR	R/W	0h	Selects one of the 16 outputs of ICL XBAR to feed into IN3 of LUTA Reset type: SYSRSn
3-1	RESERVED	R	0h	Reserved

Table 30-198. LUTCTLA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	BYPASS	R/W	1h	1 : Bypass LUT logic on PWMA 0 : PWMA driven by LUTA Reset type: SYSRSn

30.20.5.4 LUTCTLB Register (Offset = 24h) [Reset = 00000001h]

LUTCTLB is shown in Figure 30-288 and described in Table 30-199.

Return to the Summary Table.

LUT control register on PWMB

Figure 30-288. LUTCTLB Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
LUTDEC7	LUTDEC6	LUTDEC5	LUTDEC4	LUTDEC3	LUTDEC2	LUTDEC1	LUTDEC0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
SELXBAR				RESERVED			BYPASS
R/W-0h				R-0h			R/W-1h

Table 30-199. LUTCTLB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23	LUTDEC7	R/W	0h	0 : Force 0 1 : Force 1 Reset type: SYSRSn
22	LUTDEC6	R/W	0h	0 : Force 0 1 : Force 1 Reset type: SYSRSn
21	LUTDEC5	R/W	0h	0 : Force 0 1 : Force 1 Reset type: SYSRSn
20	LUTDEC4	R/W	0h	0 : Force 0 1 : Force 1 Reset type: SYSRSn
19	LUTDEC3	R/W	0h	0 : Force 0 1 : Force 1 Reset type: SYSRSn
18	LUTDEC2	R/W	0h	0 : Force 0 1 : Force 1 Reset type: SYSRSn
17	LUTDEC1	R/W	0h	0 : Force 0 1 : Force 1 Reset type: SYSRSn
16	LUTDEC0	R/W	0h	0 : Force 0 1 : Force 1 Reset type: SYSRSn
15-8	RESERVED	R	0h	Reserved
7-4	SELXBAR	R/W	0h	Selects one of the 16 outputs of ICL XBAR to feed into IN3 of LUTB Reset type: SYSRSn
3-1	RESERVED	R	0h	Reserved

Table 30-199. LUTCTLB Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	BYPASS	R/W	1h	1 : Bypass LUT logic on PWMB 0 : PWMB driven by LUTB Reset type: SYSRSn

30.20.6 HRPWMCAL_REGS Registers

Table 30-200 lists the memory-mapped registers for the HRPWMCAL_REGS registers. All register offset addresses not listed in Table 30-200 should be considered as reserved locations and the register contents should not be modified.

Table 30-200. HRPWMCAL_REGS Registers

Offset	Acronym	Register Name	Protection
42h	HRPWR	HRPWM Power Register	
4Ch	HRMSTEP	HRPWM MEP Step Register	

Complex bit access types are encoded to fit into small table cells. Table 30-201 shows the codes that are used for access types in this section.

Table 30-201. HRPWMCAL_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

30.20.6.1 HRPWR Register (Offset = 42h) [Reset = 0000h]

HRPWR is shown in Figure 30-289 and described in Table 30-202.

Return to the Summary Table.

HRPWM Power Register
This register is only accessible on EPWM modules with HRPWM capabilities.

Figure 30-289. HRPWR Register

15	14	13	12	11	10	9	8
CALPWRON	RESERVED					RESERVED	
R/W-0h	R-0-0h					R/W-0h	
7	6	5	4	3	2	1	0
RESERVED		RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	
R/W-0h		R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	

Table 30-202. HRPWR Register Field Descriptions

Bit	Field	Type	Reset	Description
15	CALPWRON	R/W	0h	MEP Calibration Power Bits (only available on ePWM1) 0: Disables MEP calibration logic in the HRPWM and reduces power consumption. 1: Enables MEP calibration logic Reset type: SYSRSn
14-10	RESERVED	R-0	0h	Reserved
9-6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	RESERVED	R	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1-0	RESERVED	R/W	0h	Reserved

30.20.6.2 HRMSTEP Register (Offset = 4Ch) [Reset = 0000h]

HRMSTEP is shown in Figure 30-290 and described in Table 30-203.

Return to the Summary Table.

HRPWM MEP Step Register
This register is only accessible on EPWM modules with HRPWM capabilities. Only 16 bit accesses are allowed for this register. Debugger access in 32 bit mode may display incorrect values.

Figure 30-290. HRMSTEP Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
HRMSTEP							
R/W-0h							

Table 30-203. HRMSTEP Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R-0	0h	Reserved
7-0	HRMSTEP	R/W	0h	High Resolution MEP Step When auto-conversion is enabled (HRCNFG[AUTOCONV] = 1), This 8-bit field contains the MEP_ScaleFactor (number of MEP steps per coarse steps) used by the hardware to automatically convert the value in the CMPAHR, CMPBHR, DBFEDHR, DBREDHR , TBPHSHR, or TBPRDHR register to a scaled micro-edge delay on the high-resolution ePWM output. The value in this register is written by the SFO calibration software at the end of each calibration run. Reset type: SYSRSn

Chapter 31

Enhanced Quadrature Encoder Pulse (eQEP)

The enhanced Quadrature Encoder Pulse (eQEP) module described here is a Type 2 eQEP. See the *C2000 Real-Time Control Peripheral Reference Guide* for a list of all devices with a module of the same type to determine the differences between types and for a list of device-specific differences within a type.

The enhanced quadrature encoder pulse (eQEP) module is used for direct interface with a linear or rotary incremental encoder to get position, direction, and speed information from a rotating machine for use in a high-performance motion and position-control system.

31.1 Introduction

An incremental encoder disk is patterned with a track of slots along the periphery, as shown in Figure 31-1. These slots create an alternating pattern of dark and light lines. The disk count is defined as the number of dark and light line pairs that occur per revolution (lines per revolution). As a rule, a second track is added to generate a signal that occurs once per revolution (index signal: QEPI), which can be used to indicate an absolute position. Encoder manufacturers identify the index pulse using different terms such as index, marker, home position, and zero reference

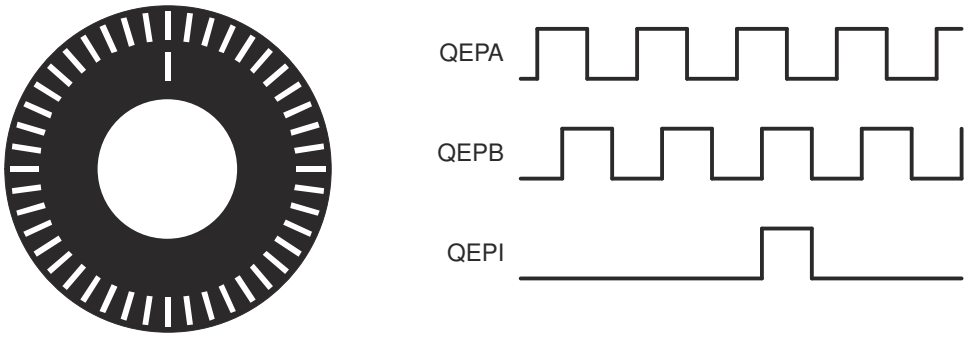
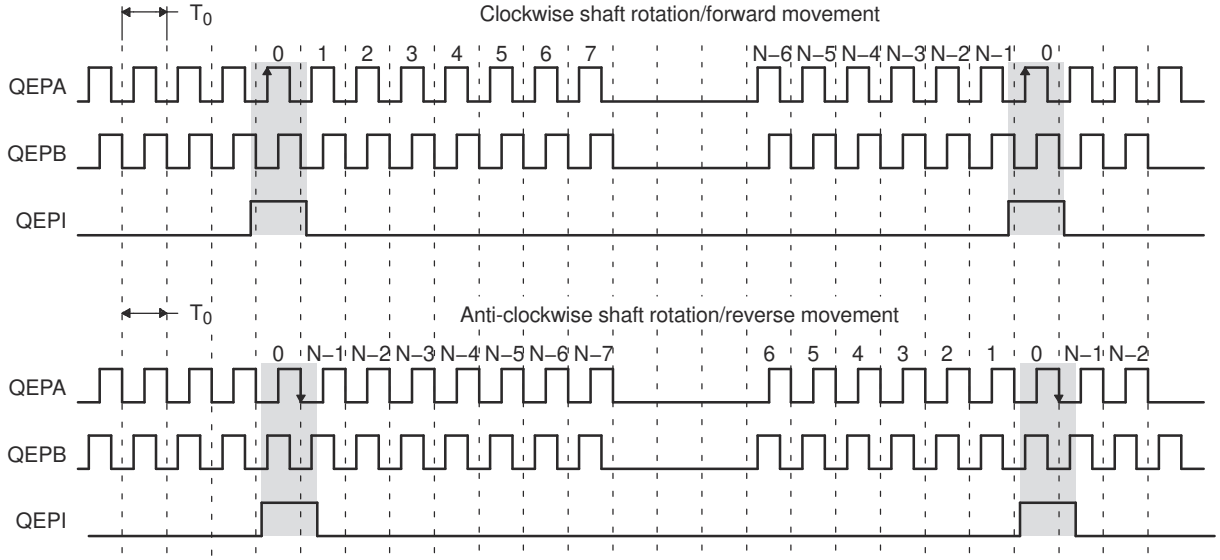


Figure 31-1. Optical Encoder Disk

To derive direction information, the lines on the disk are read out by two different photo-elements that "look" at the disk pattern with a mechanical shift of 1/4 the pitch of a line pair between them. This shift is detected with a reticle or mask that restricts the view of the photo-element to the desired part of the disk lines. As the disk rotates, the two photo-elements generate signals that are shifted 90° out of phase from each other. These are commonly called the quadrature QEPA and QEPB signals. The clockwise direction for most encoders is defined as the QEPA channel going positive before the QEPB channel and conversely, as shown in Figure 31-2.



Legend: N = lines per revolution

Figure 31-2. QEP Encoder Output Signal for Forward/Reverse Movement

The encoder wheel typically makes one revolution for every revolution of the motor, or the wheel can be at a geared rotation ratio with respect to the motor. Therefore, the frequency of the digital signal coming from the QEPA and QEPB outputs varies proportionally with the velocity of the motor. For example, a 2000-line encoder

directly coupled to a motor running at 5000 revolutions-per-minute (rpm) results in a frequency of 166.6kHz, so by measuring the frequency of either the QEPA or QEPB output, the processor can determine the velocity of the motor.

Quadrature encoders from different manufacturers come with two forms of index pulse (gated index pulse or ungated index pulse) as shown in Figure 31-3. A nonstandard form of index pulse is ungated. In the ungated configuration, the index edges are not necessarily coincident with A and B signals. The gated index pulse is aligned to any of the four quadrature edges and width of the index pulse and can be equal to a quarter, half, or full period of the quadrature signal.

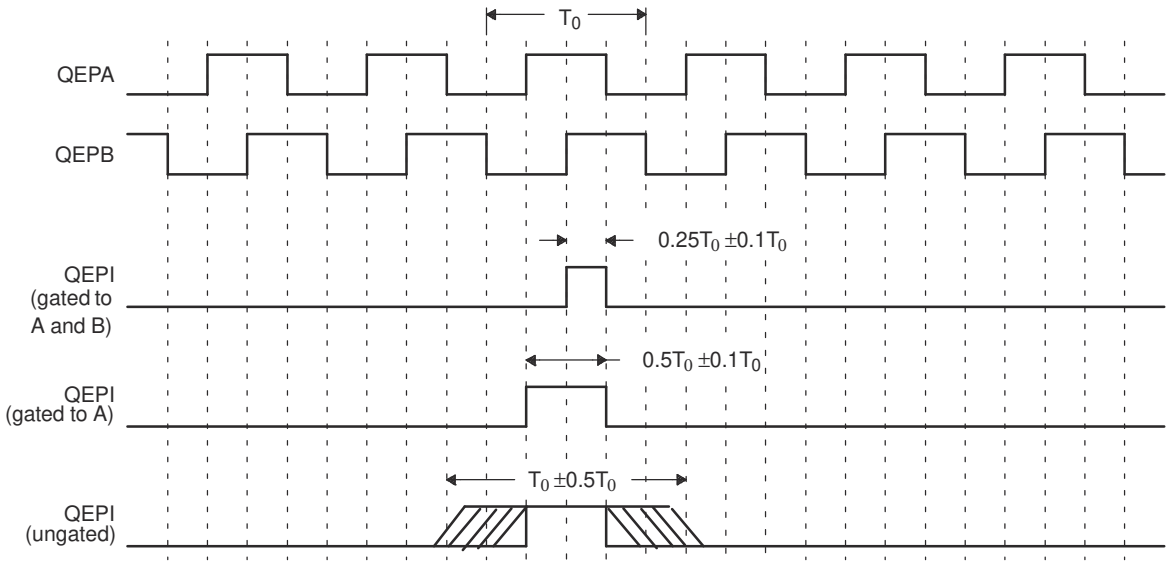


Figure 31-3. Index Pulse Example

Some typical applications of shaft encoders include robotics and computer input in the form of a mouse. Inside your mouse you can see where the mouse ball spins a pair of axles (a left/right, and an up/down axle). These axles are connected to optical shaft encoders that effectively tell the computer how fast and in what direction the mouse is moving.

General Issues: Estimating velocity from a digital position sensor is a cost-effective strategy in motor control. Two different first order approximations for velocity can be written as:

$$v(k) \approx \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T} \quad (24)$$

$$v(k) \approx \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \quad (25)$$

where:

- v(k) = Velocity at time instant k
- x(k) = Position at time instant k
- x(k-1) = Position at time instant k-1
- T = Fixed unit time or inverse of velocity calculation rate
- ΔX = Incremental position movement in unit time
- t(k) = Time instant "k"
- t(k-1) = Time instant "k-1"
- X = Fixed unit position
- ΔT = Incremental time elapsed for unit position movement

Equation 24 is the conventional approach to velocity estimation and requires a time base to provide a unit time event for velocity calculation. Unit time is basically the inverse of the velocity calculation rate.

The encoder count (position) is read once during each unit time event. The quantity [x(k) - x(k-1)] is formed by subtracting the previous reading from the current reading. Then the velocity estimate is computed by multiplying by the known constant 1/T (where T is the constant time between unit time events and is known in advance).

Estimation based on Equation 24 has an inherent accuracy limit directly related to the resolution of the position sensor and the unit time period T. For example, consider a 500 line-per-revolution quadrature encoder with a velocity calculation rate of 400Hz. When used for position, the quadrature encoder gives a four-fold increase in resolution; in this case, 2000 counts-per-revolution. The minimum rotation that can be detected is, therefore, 0.0005 revolutions, which gives a velocity resolution of 12rpm when sampled at 400Hz. While this resolution can be satisfactory at moderate or high speeds, for example 1% error at 1200rpm, this resolution clearly proves inadequate at low speeds. In fact, at speeds below 12rpm, the speed estimate is erroneously zero much of the time.

At low speed, Equation 25 provides a more accurate approach. It requires a position sensor that outputs a fixed interval pulse train, such as the aforementioned quadrature encoder. The width of each pulse is defined by motor speed for a given sensor resolution. Equation 25 can be used to calculate motor speed by measuring the elapsed time between successive quadrature pulse edges. However, this method suffers from the opposite limitation, as does Equation 24. A combination of relatively large motor speeds and high sensor resolution makes the time interval ΔT small, and thus more greatly influenced by the timer resolution. This can introduce considerable error into high-speed estimates.

For systems with a large speed range (that is, speed estimation is needed at both low and high speeds), one approach is to use Equation 25 at low speed and have the DSP software switch over to Equation 24 when the motor speed rises above some specified threshold.

31.1.1 EQEP Related Collateral

Foundational Materials

- C28x Academy - EQEP
- C29x Academy - Enhanced Quadrature Encoder Pulse (EQEP)
- Interfacing with Quadrature Encoders (Video)
- *Real-Time Control Reference Guide*
  - Refer to the Encoders section

Getting Started Materials

- *C2000™ Position Manager PTO API Reference Guide Application Report*

Expert Materials

- *CW/CCW Support on the C2000 eQEP Module Application Report*

31.2 Configuring Device Pins

The GPIO mux registers must be configured to connect this peripheral to the device pins. To avoid glitches on the pins, the GPyGMUX bits must be configured first (while keeping the corresponding GPyMUX bits at the default of zero), followed by writing the GPyMUX register to the desired value.

For proper operation of the eQEP module, input GPIO pins must be configured using the GPxQSELn registers for synchronous input mode (with or without qualification). The asynchronous mode cannot be used for eQEP input pins. The internal pullups can be configured in the GPyPUD register.

See the *General-Purpose Input/Output (GPIO)* chapter for more details on GPIO mux and settings.

31.3 Description

This section provides the eQEP inputs, memory map, and functional description.

31.3.1 EQEP Inputs

The eQEP inputs include two pins for quadrature-clock mode or direction-count mode, an index (or 0 marker), and a strobe input. The eQEP module requires that the QEPA, QEPB, and QEPI inputs are synchronized to SYSCLK prior to entering the module. The application code can enable the synchronous GPIO input feature on any eQEP-enabled GPIO pins (see the *General-Purpose Input/Output (GPIO)* chapter for more details).

- **QEPA/XCLK and QEPB/XDIR:** These two pins can be used in quadrature-clock mode or direction-count mode.
  - Quadrature-clock Mode: The eQEP encoders provide two square wave signals (A and B) 90 electrical degrees out of phase. This phase relationship is used to determine the direction of rotation of the input shaft and number of eQEP pulses from the index position to derive the relative position information. For forward or clockwise rotation, QEPA signal leads QEPB signal and conversely. The quadrature decoder uses these two inputs to generate quadrature-clock and direction signals.
  - Direction-count Mode: In direction-count mode, direction and clock signals are provided directly from the external source. Some position encoders have this type of output instead of quadrature output. The QEPA pin provides the clock input and the QEPB pin provides the direction input.
- **QEPI: Index or Zero Marker:** The eQEP encoder uses an index signal to assign an absolute start position from which position information is incrementally encoded using quadrature pulses. This pin is connected to the index output of the eQEP encoder to optionally reset the position counter for each revolution. This signal can be used to initialize or latch the position counter on the occurrence of a desired event on the index pin.
- **QEPS: Strobe Input:** This general-purpose strobe signal can initialize or latch the position counter on the occurrence of a desired event on the strobe pin. This signal is typically connected to a sensor or limit switch to notify that the motor has reached a defined position.

Input signals to the eQEP (QEPA, QEPB, QEPI and QEPS) can come from multiple sources; that is, device pin, CMPSSx, or PWMXBARx. One typical use case is if SinCos transducers are used in the motor control system to estimate the position of motor shaft and Index signal is coming from traditional rotary encoder, source of the eQEP signals (QEPA, QEPB and QEPI) can be configured as output of CMPSSx that decodes the Sin, Cos, and Index signals. Figure 31-4 illustrates the use case.

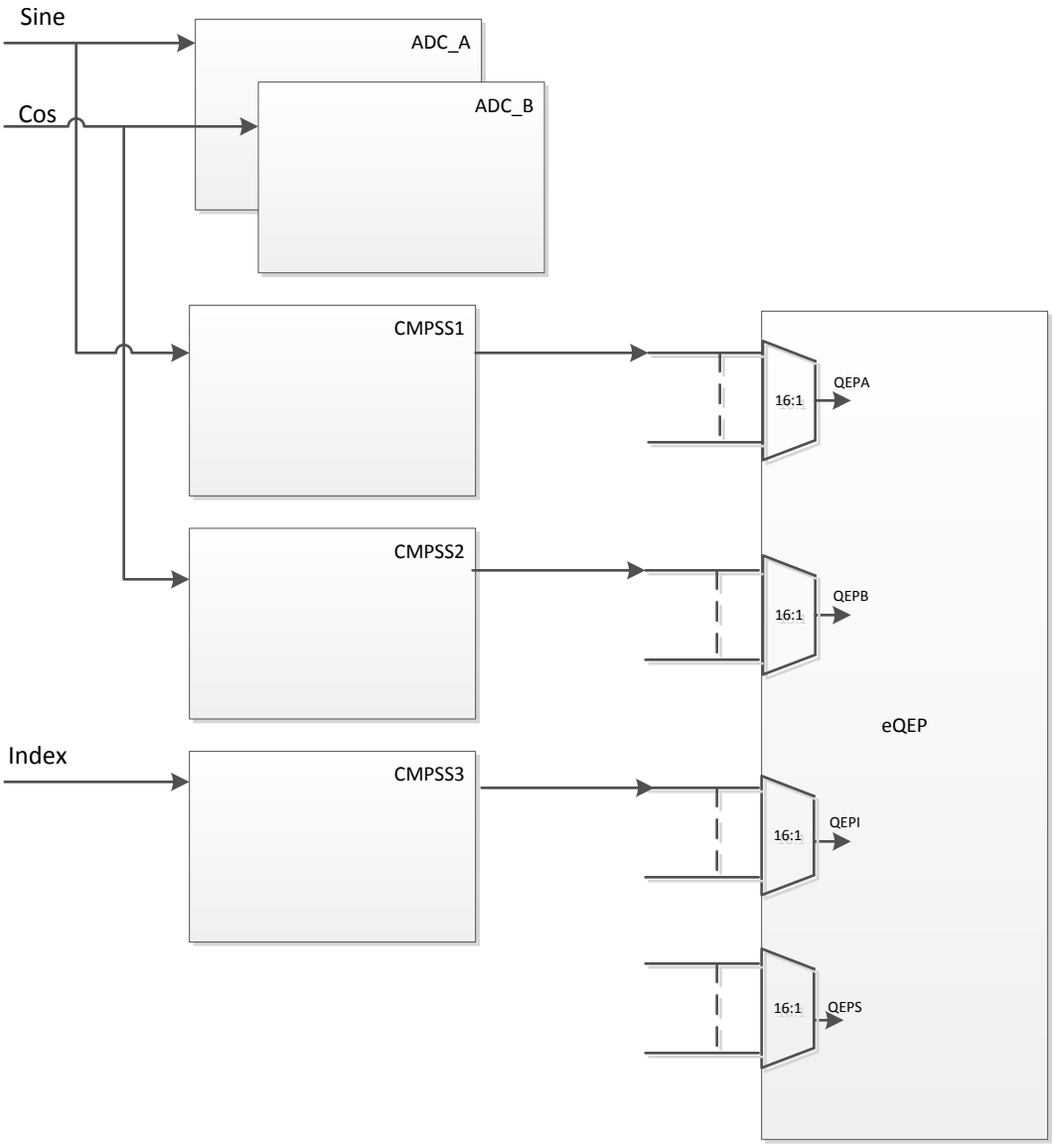


Figure 31-4. Using eQEP to Decode Signals from SinCos Transducer

Note

Configuration of QEPSRCSEL register to select the source of QEPA, QEPB, and QEPI signals can lead to unexpected transition on these signals, which can cause an undesirable outcome if eQEP is already running. Make sure that the eQEP is disabled before configuring the QEPSRCSEL register for input signals.

31.3.2 Functional Description

The eQEP peripheral contains the following major functional units (as shown in Figure 31-5):

- Programmable input qualification for each pin (part of the GPIO MUX)
- Quadrature decoder unit (QDU)
- Position counter and control unit for position measurement (PCCU)
- Quadrature edge-capture unit for low-speed measurement (QCAP)
- Unit time base for speed/frequency measurement (UTIME)
- Watchdog timer for detecting stalls (QWDOG)
- Quadrature Mode Adapter (QMA)

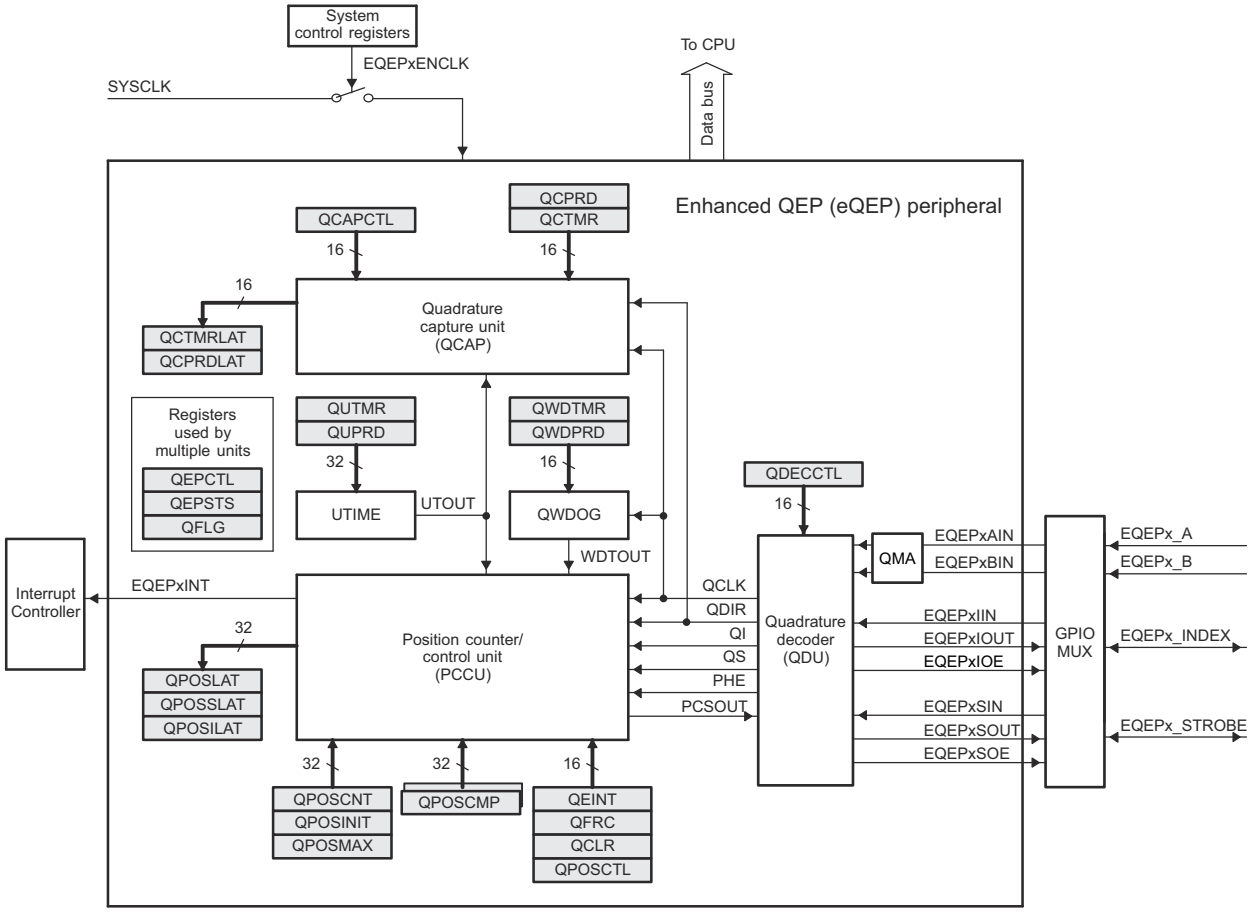


Figure 31-5. Functional Block Diagram of the eQEP Peripheral

31.3.3 eQEP Memory Map

Table 31-1 lists the registers with the memory locations, sizes, and reset values.

Table 31-1. EQEP Memory Map

Name	Offset	Size(x16)/ #shadow	Reset	Register Description
QPOSCNT	0x00	2/0	0x0000 0000	eQEP Position Counter
QPOSINIT	0x02	2/0	0x0000 0000	eQEP Initialization Position Count
QPOSMAX	0x04	2/0	0x0000 0000	eQEP Maximum Position Count
QPOSCMP	0x06	2/1	0x0000 0000	eQEP Position-compare
QPOSILAT	0x08	2/0	0x0000 0000	eQEP Index Position Latch
QPOSSLAT	0x0A	2/0	0x0000 0000	eQEP Strobe Position Latch
QPOSLAT	0x0C	2/0	0x0000 0000	eQEP Position Latch
QUTMR	0x0E	2/0	0x0000 0000	eQEP Unit Timer
QUPRD	0x10	2/0	0x0000 0000	eQEP Unit Period Register
QWDTMR	0x12	1/0	0x0000	eQEP Watchdog Timer
QWDPRD	0x13	1/0	0x0000	eQEP Watchdog Period Register
QDECCTL	0x14	1/0	0x0000	eQEP Decoder Control Register
QEPCTL	0x15	1/0	0x0000	eQEP Control Register
QCAPCTL	0x16	1/0	0x0000	eQEP Capture Control Register
QPOSCTL	0x17	1/0	0x0000	eQEP Position-compare Control Register
QEINT	0x18	1/0	0x0000	eQEP Interrupt Enable Register
QFLG	0x19	1/0	0x0000	eQEP Interrupt Flag Register
QCLR	0x1A	1/0	0x0000	eQEP Interrupt Clear Register
QFRC	0x1B	1/0	0x0000	eQEP Interrupt Force Register
QEPSTS	0x1C	1/0	0x0000	eQEP Status Register
QCTMR	0x1D	1/0	0x0000	eQEP Capture Timer
QCPRD	0x1E	1/0	0x0000	eQEP Capture Period Register
QCTMRLAT	0x1F	1/0	0x0000	eQEP Capture Timer Latch
QCPRDLAT	0x20	1/0	0x0000	eQEP Capture Period Latch
Reserved	0x21 to 0x2F	15/0		
REV	0x30	2/0	0x0000	eQEP Revision Number
QEPSTROBESEL	0x32	2/0	0x0000	eQEP Strobe select register
QMACTRL	0x34	2/0	0x0000	eQEP QMA Control register
QEPSRCSEL	0x36	2/0	0x0000	eQEP Source Select Register
Reserved	0x38 to 0x3F	8/0		

31.4 Quadrature Decoder Unit (QDU)

Figure 31-6 shows a functional block diagram of the QDU.

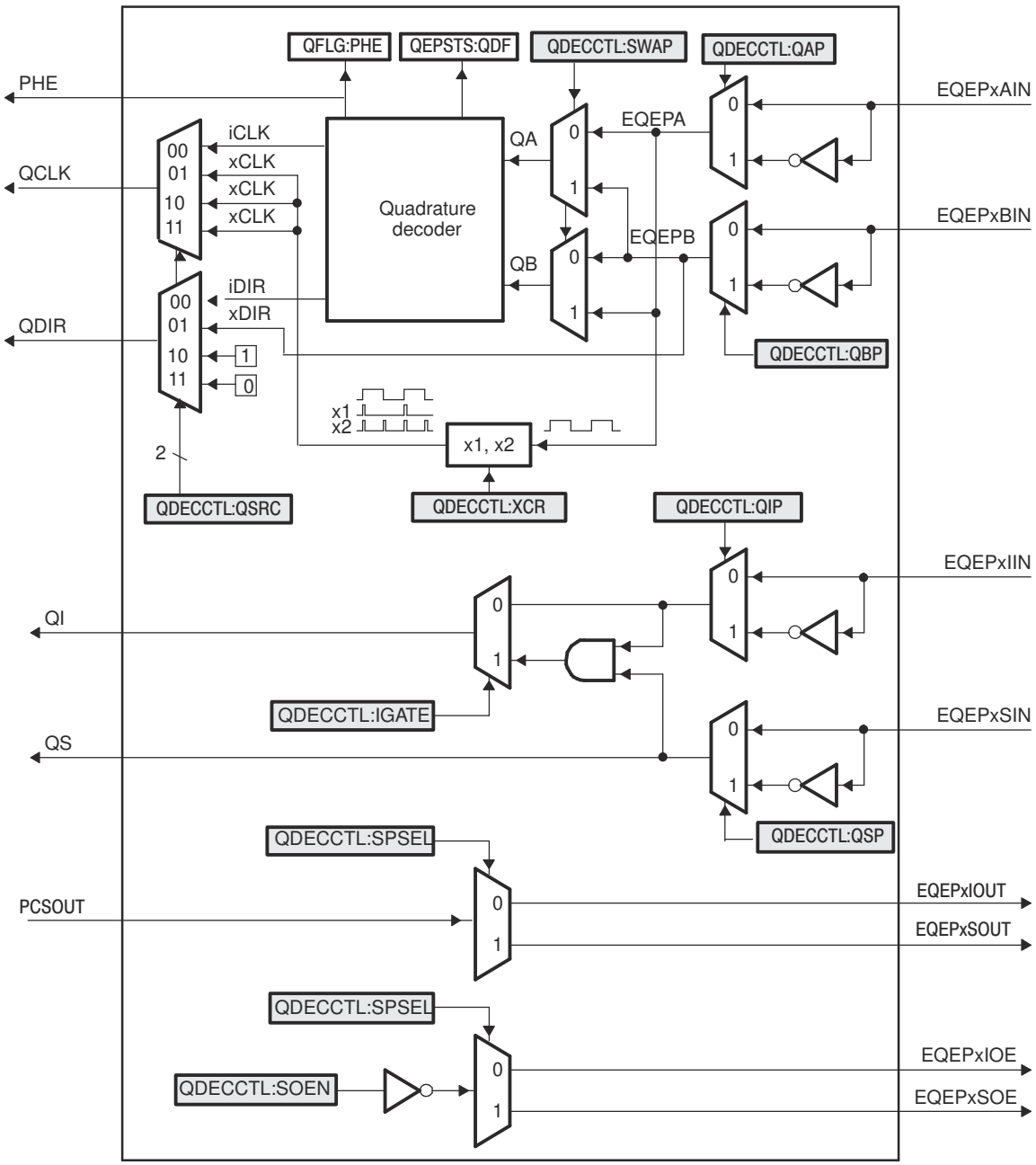


Figure 31-6. Functional Block Diagram of Decoder Unit

31.4.1 Position Counter Input Modes

Clock and direction input to the position counter is selected using QDECCTL[QSRC] bits, based on interface input requirement as follows:

- Quadrature-count mode
- Direction-count mode
- UP-count mode
- DOWN-count mode

31.4.1.1 Quadrature Count Mode

The quadrature decoder generates the direction and clock to the position counter in quadrature count mode.

Direction Decoding

The direction decoding logic of the eQEP circuit determines which one of the sequences (QEPA, QEPB) is the leading sequence and accordingly updates the direction information in the QEPSTS[QDF] bit. Table 31-2 and Figure 31-7 show the direction decoding logic in truth table and state machine form. Both edges of the QEPA and QEPB signals are sensed to generate count pulses for the position counter. Therefore, the frequency of the clock generated by the eQEP logic is four times that of each input sequence. Figure 31-8 shows the direction decoding and clock generation from the eQEP input signals.

Table 31-2. Quadrature Decoder Truth Table

Previous Edge	Present Edge	QDIR	QPOSCNT
QA↑	QB↑	UP	Increment
	QB↓	DOWN	Decrement
	QA↓	TOGGLE	Increment or Decrement
QA↓	QB↓	UP	Increment
	QB↑	DOWN	Decrement
	QA↑	TOGGLE	Increment or Decrement
QB↑	QA↑	DOWN	Decrement
	QA↓	UP	Increment
	QB↓	TOGGLE	Increment or Decrement
QB↓	QA↓	DOWN	Decrement
	QA↑	UP	Increment
	QB↑	TOGGLE	Increment or Decrement

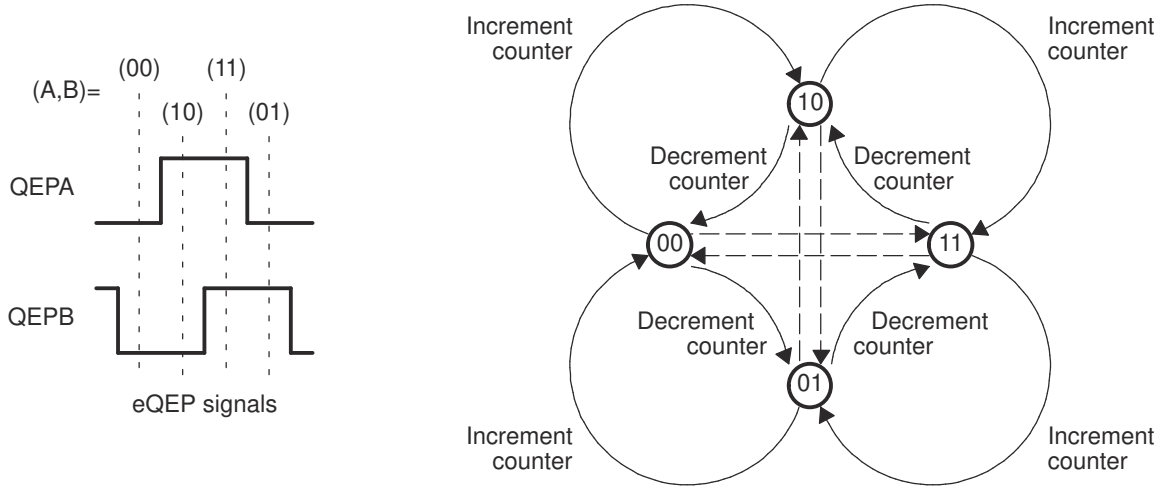


Figure 31-7. Quadrature Decoder State Machine

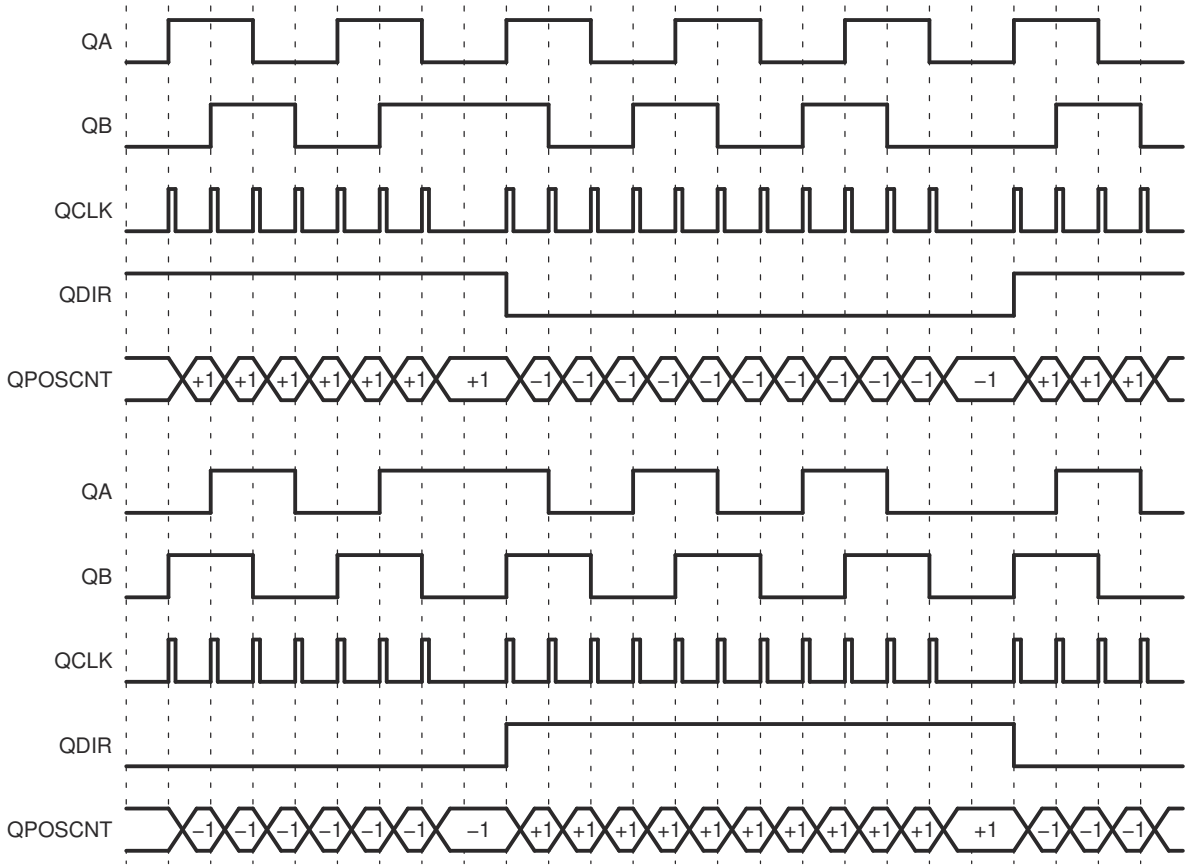


Figure 31-8. Quadrature-clock and Direction Decoding

Phase Error Flag In normal operating conditions, quadrature inputs QEPA and QEPB is 90 degrees out of phase. The phase error flag (PHE) is set in the QFLG register and the QPOSCNT value can be incorrect and offset by multiples of 1 or 3. That is, when edge transition is detected simultaneously on the QEPA and QEPB signals to optionally generate interrupts. State transitions marked by dashed lines in Figure 31-7 are invalid transitions that generate a phase error.

Count Multiplication The eQEP position counter provides 4x times the resolution of an input clock by generating a quadrature-clock (QCLK) on the rising/falling edges of both eQEP input clocks (QEPA and QEPB) as shown in Figure 31-8.

Reverse Count In normal quadrature count operation, QEPA input is applied to the QA input of the quadrature decoder and the QEPB input is applied to the QB input of the quadrature decoder. Reverse counting is enabled by setting the SWAP bit in the QDECCTL register. This swaps the input to the quadrature decoder; thereby, reversing the counting direction.

31.4.1.2 Direction-Count Mode

Some position encoders provide direction and clock outputs, instead of quadrature outputs. In such cases, direction-count mode can be used. The QEPA input provides the clock for the position counter and the QEPB input has the direction information. The position counter is incremented on every rising edge of a QEPA input when the direction input is high, and decremented when the direction input is low.

31.4.1.3 Up-Count Mode

The counter direction signal is hard-wired for up-count and the position counter is used to measure the frequency of the QEPA input. Clearing the QDECCTL[XCR] bit enables clock generation to the position counter on both edges of the QEPA input; thereby, increasing the measurement resolution by a factor of 2x. In up-count mode, we recommend that the application not configure QEPB as a GPIO mux option, or make sure that a signal edge is not generated on the QEPB input.

31.4.1.4 Down-Count Mode

The counter direction signal is hardwired for a down-count and the position counter is used to measure the frequency of the QEPA input. Clearing the QDECCTL[XCR] bit enables clock generation to the position counter on both edges of a QEPA input, thereby increasing the measurement resolution by a factor of 2x. In down-count mode, the application must not configure QEPB as a GPIO mux option or make sure that a signal edge is not generated on the QEPB input.

31.4.2 eQEP Input Polarity Selection

Each eQEP input can be inverted using QDECCTL[8:5] control bits. As an example, setting the QDECCTL[QIP] bit inverts the index input.

31.4.3 Position-Compare Sync Output

The enhanced eQEP peripheral includes a position-compare unit that is used to generate the position-compare sync signal on compare match between the position-counter register (QPOSCNT) and the position- compare register (QPOSCMP). This sync signal can be output using an index pin or strobe pin of the EQEP peripheral.

Setting the QDECCTL[SOEN] bit enables the position-compare sync output and the QDECCTL[SPSEL] bit selects either an eQEP index pin or an eQEP strobe pin.

31.5 Position Counter and Control Unit (PCCU)

The position-counter and control unit provides two configuration registers (QEPCTL and QPOSCTL) for setting up position-counter operational modes, position-counter initialization/latch modes and position-compare logic for sync signal generation.

31.5.1 Position Counter Operating Modes

Position-counter data can be captured in different manners. In some systems, the position counter is accumulated continuously for multiple revolutions and the position-counter value provides the position information with respect to the known reference. An example of this is the quadrature encoder mounted on the motor controlling the print head in the printer. Here the position counter is reset by moving the print head to the home position and then the position counter provides absolute position information with respect to home position.

In other systems, the position counter is reset on every revolution using index pulse, and the position counter provides a rotor angle with respect to the index pulse position.

The position counter can be configured to operate in following four modes

- Position-Counter Reset on Index Event
- Position-Counter Reset on Maximum Position
- Position-Counter Reset on the first Index Event
- Position-Counter Reset on Unit Time Out Event (Frequency Measurement)

In all the above operating modes, the position counter is reset to 0 on overflow and to the QPOSMAX register value on underflow. Overflow occurs when the position counter counts up after the QPOSMAX value. Underflow occurs when the position counter counts down after 0. The Interrupt flag is set to indicate overflow/underflow in QFLG register.

31.5.1.1 Position Counter Reset on Index Event (QEPCTL[PCRM] = 00)

If the index event occurs during the forward movement, then the position counter is reset to 0 on the next eQEP clock. If the index event occurs during the reverse movement, then the position counter is reset to the value in the QPOSMAX register on the next eQEP clock.

The first index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (QEPSTS[FIMF]) and direction on the first index event marker (QEPSTS[FIDF]) in QEPSTS registers, the eQEP peripheral also remembers the quadrature edge on the first index marker so that same relative quadrature transition is used for index event reset operation.

For example, if the first reset operation occurs on the falling edge of QEPB during the forward direction, then all the subsequent reset must be aligned with the falling edge of QEPB for the forward rotation and on the rising edge of QEPB for the reverse rotation as shown in Figure 31-9.

The position-counter value is latched to the QPOSILAT register and direction information is recorded in the QEPSTS[QDLF] bit on every index event marker. The position-counter error flag (QEPSTS[PCEF]) and error interrupt flag (QFLG[PCE]) are set if the latched value is not equal to 0 or QPOSMAX. The position-counter error flag (QEPSTS[PCEF]) is updated on every index event marker and an interrupt flag (QFLG[PCE]) is set on error that can be cleared only through software.

The index event latch configuration QEPCTL[IEL] must be configured to 00 or 11 when pcrm = 0 and the position counter error flag/interrupt flag are generated only in index event reset mode. The position counter value is latched into the IPOSLAT register on every index marker.

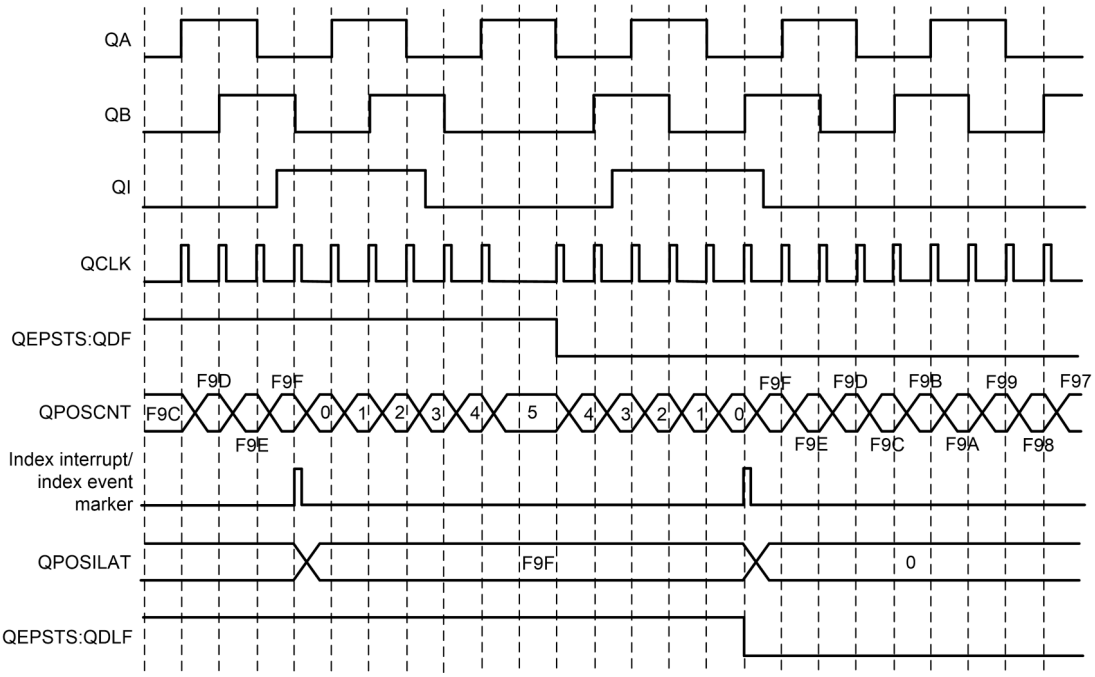


Figure 31-9. Position Counter Reset by Index Pulse for 1000-Line Encoder (QPOSMAX = 3999 or 0xF9F)

Note

In case of a boundary condition where the time period between the Index Event and the previous QCLK edge is less than SYSCLK period, then QPOSCNT gets reset to zero or QPOSMAX in the same SYSCLK cycle and does not wait for the next QCLK edge to occur.

31.5.1.2 Position Counter Reset on Maximum Position (QEPCTL[PCRM] = 01)

If the position counter is equal to QPOSMAX, then the position counter is reset to 0 on the next eQEP clock for forward movement and position counter overflow flag is set. If the position counter is equal to ZERO, then the position counter is reset to QPOSMAX on the next QEP clock for reverse movement and position-counter underflow flag is set. Figure 31-10 shows the position-counter reset operation in this mode.

The first index marker fields (QEPSTS[FIDF] and QEPSTS[FIMF]) are not applicable in this mode.

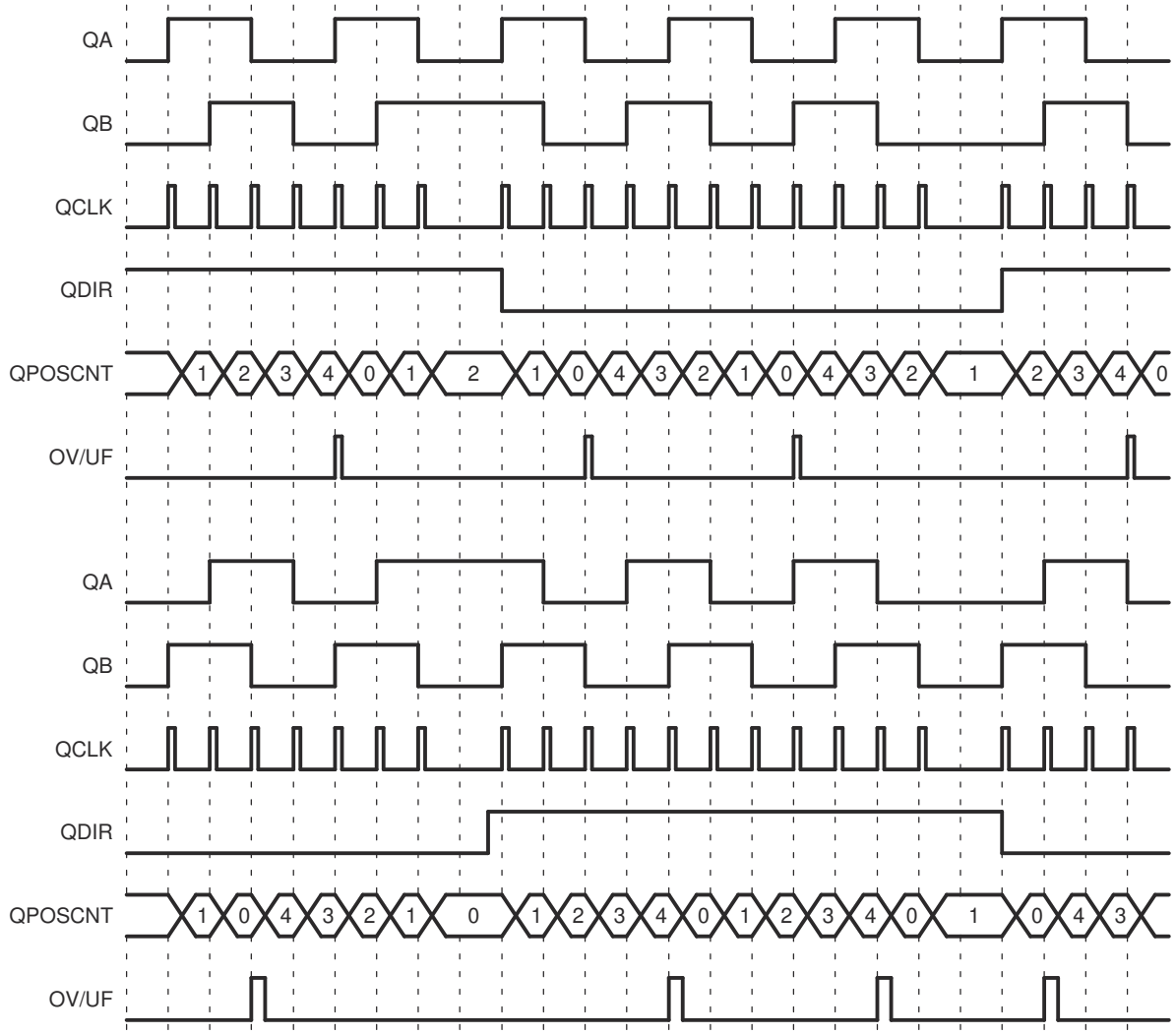


Figure 31-10. Position Counter Underflow/Overflow (QPOSMAX = 4)

31.5.1.3 Position Counter Reset on the First Index Event (QEPCTL[PCRM] = 10)

If the index event occurs during forward movement, then the position counter is reset to 0 on the next eQEP clock. If the index event occurs during the reverse movement, then the position counter is reset to the value in the QPOSMAX register on the next eQEP clock. Note that this is done only on the first occurrence and subsequently the position-counter value is not reset on an index event; rather, the position-counter value is reset based on the maximum position as described in Section 31.5.1.2.

The first index marker fields (QEPSTS[FIDF] and QEPSTS[FIMF]) are not applicable in this mode.

31.5.1.4 Position Counter Reset on Unit Time-out Event (QEPCTL[PCRM] = 11)

In this mode, QPOSCNT is set to 0 or QPOMAX, depending on the direction mode selected by QDECCTL[QSRC] bits on a unit time event. This is useful for frequency measurement.

31.5.2 Position Counter Latch

The eQEP index and strobe input can be configured to latch the position counter (QPOSCNT) into QPOSILAT and QPOSSLAT, respectively, on occurrence of a definite event on these pins.

31.5.2.1 Index Event Latch

In some applications, it is not desirable to reset the position counter on every index event and instead it can be required to operate the position counter in full 32-bit mode (QEPCTL[PCRM] = 01 and QEPCTL[PCRM] = 10 modes).

In such cases, the eQEP position counter can be configured to latch on the following events and direction information is recorded in the QEPSTS[QDLF] bit on every index event marker.

- Latch on Rising edge (QEPCTL[IEL] = 01)
- Latch on Falling edge (QEPCTL[IEL] = 10)
- Latch on Index Event Marker (QEPCTL[IEL] = 11)

This is particularly useful as an error checking mechanism to check if the position counter accumulated the correct number of counts between index events. As an example, the 1000-line encoder must count 4000 times when moving in the same direction between the index events.

The index event latch interrupt flag (QFLG[IEL]) is set when the position counter is latched to the QPOSILAT register. The index event latch configuration bits (QEPCTL[IEL]) are ignored when QEPCTL[PCRM] = 00.

Latch on Rising Edge (QEPCTL[IEL] = 01)	The position-counter value (QPOSCNT) is latched to the QPOSILAT register on every rising edge of an index input.
Latch on Falling Edge (QEPCTL[IEL] = 10)	The position-counter value (QPOSCNT) is latched to the QPOSILAT register on every falling edge of index input.
Latch on Index Event Marker/Software Index Marker (QEPCTL[IEL] = 11	The first index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (QEPSTS[FIMF]) and the direction on the first index event marker (QEPSTS[FIDF]) in the QEPSTS registers. The eQEP peripheral also remembers the quadrature edge on the first index marker so that the same relative quadrature transition is used for latching the position counter (QEPCTL[IEL] = 11).

Figure 31-11 shows the position counter latch using an index event marker.

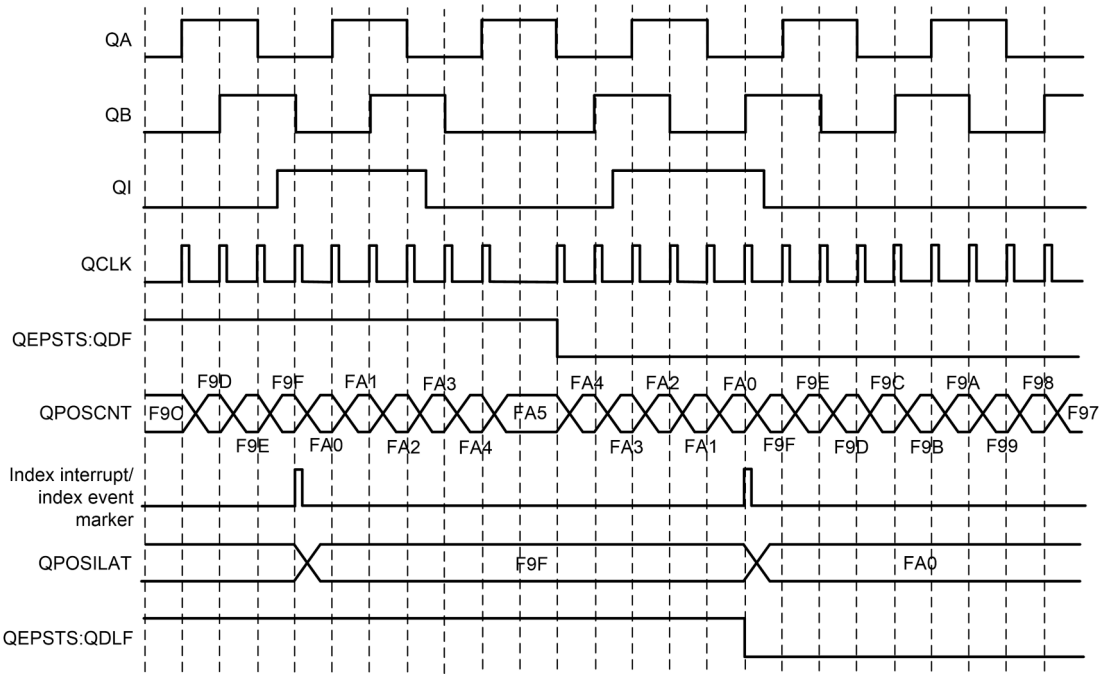


Figure 31-11. Software Index Marker for 1000-line Encoder (QEPCTL[IEL] = 1)

31.5.2.2 Strobe Event Latch

The position-counter value is latched to the QPOSSLAT register on the rising edge of the strobe input by clearing the QEPCTL[SEL] bit.

If the QEPCTL[SEL] bit is set, then the position-counter value is latched to the QPOSSLAT register on the rising edge of the strobe input for forward direction, and on the falling edge of the strobe input for reverse direction as shown in Figure 31-12.

The strobe event latch interrupt flag (QFLG[SEL) is set when the position counter is latched to the QPOSSLAT register.

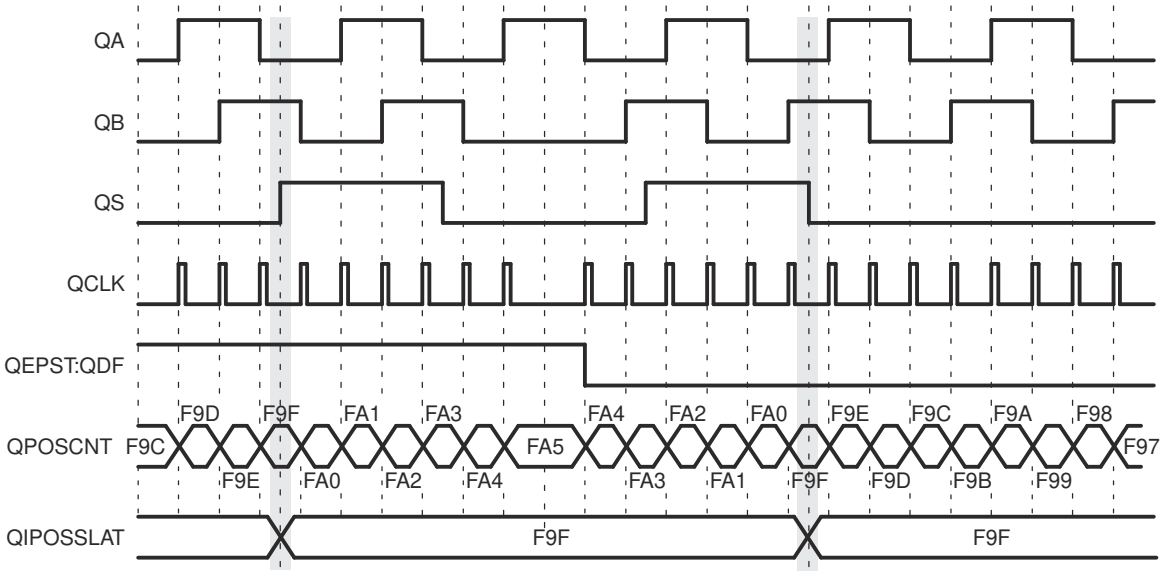


Figure 31-12. Strobe Event Latch (QEPCTL[SEL] = 1)

There is an added feature on Type 2.0 eQEP where position-counter value can also be latched on ADCSOCA and ADCSOSCB events by configuring the register QEPSTROBESEL.STROBESEL as shown in Figure 31-13. To use the ADCSOCA/B events for the QS signal, configuration of the QEPSRCSEL.QEPSSEL to be non-zero is needed..

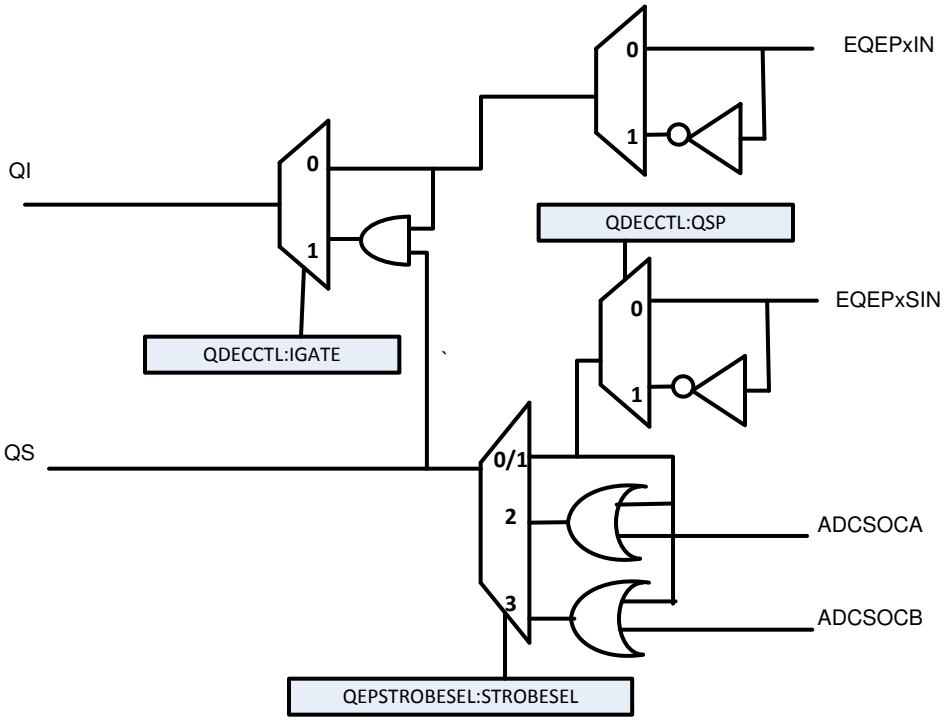


Figure 31-13. Latching Position Counter on ADCSOCA/ADCSOCB Event

31.5.3 Position Counter Initialization

The position counter can be initialized using the following events:

- Index event
- Strobe event
- Software initialization

Index Event Initialization (IEI)	The QEPI index input can be used to trigger the initialization of the position counter at the rising or falling edge of the index input. If the QEPCTL[IEI] bits are 10, then the position counter (QPOSCNT) is initialized with a value in the QPOSINIT register on the rising edge of index input. Conversely, if the QEPCTL[IEI] bits are 11, initialization is on the falling edge of the index input.
Strobe Event Initialization (SEI)	If the QEPCTL[SEI] bits are 10, then the position counter is initialized with a value in the QPOSINIT register on the rising edge of strobe input. If QEPCTL[SEL] bits are 11, then the position counter is initialized with a value in the QPOSINIT register on the rising edge of strobe input for forward direction and on the falling edge of strobe input for reverse direction.
Software Initialization (SWI)	The position counter can be initialized in software by writing a 1 to the QEPCTL[SWI] bit. This bit is not automatically cleared. While the bit is still set, if a 1 is written to the bit again, the position counter is re-initialized.

31.5.4 eQEP Position-compare Unit

The eQEP peripheral includes a position-compare unit that is used to generate a sync output and interrupt on a position-compare match. Figure 31-14 shows a diagram. The position-compare (QPOSCMP) register is shadowed and shadow mode can be enabled or disabled using the QPOSCTL[PSSHDW] bit. If the shadow mode is not enabled, the CPU writes directly to the active position compare register.

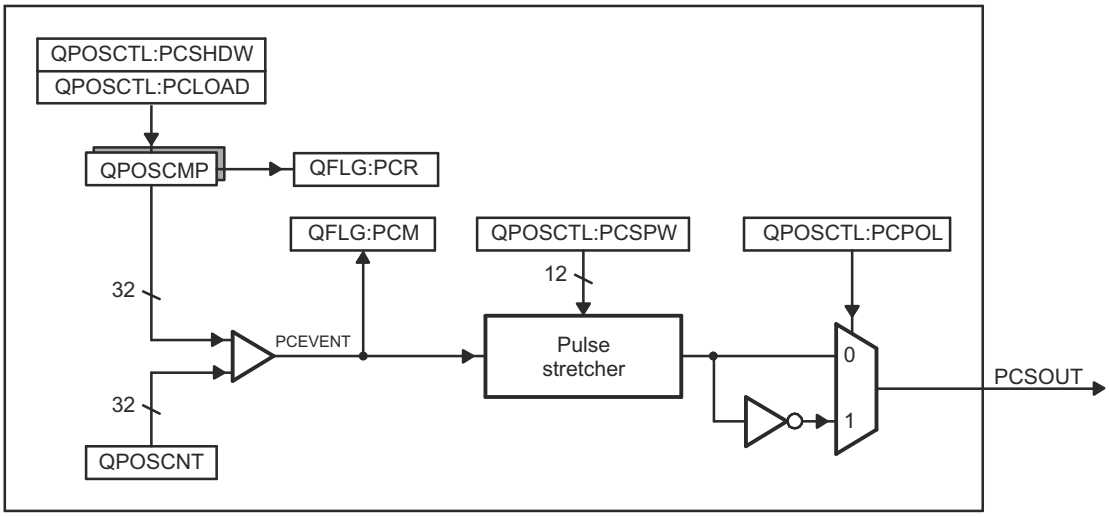


Figure 31-14. eQEP Position-compare Unit

In shadow mode, you can configure the position-compare unit (QPOSCTL[PCLOAD]) to load the shadow register value into the active register on the following events, and to generate the position-compare ready (QFLG[PCR]) interrupt after loading.

- Load on compare match
- Load on position-counter zero event

The position-compare match (QFLG[PCM]) is set when the position-counter value (QPOSCNT) matches with the active position-compare register (QPOSCMP) and the position-compare sync output of the programmable pulse width is generated on compare-match to trigger an external device.

For example, if QPOSCMP = 2, the position-compare unit generates a position-compare event on 1 to 2 transitions of the eQEP position counter for forward counting direction and on 3 to 2 transitions of the eQEP position counter for reverse counting direction (see Figure 31-15).

See the register section for the layout of the eQEP Position-Compare Control Register (QPOSCTL) and description of the QPOSCTL bit fields.

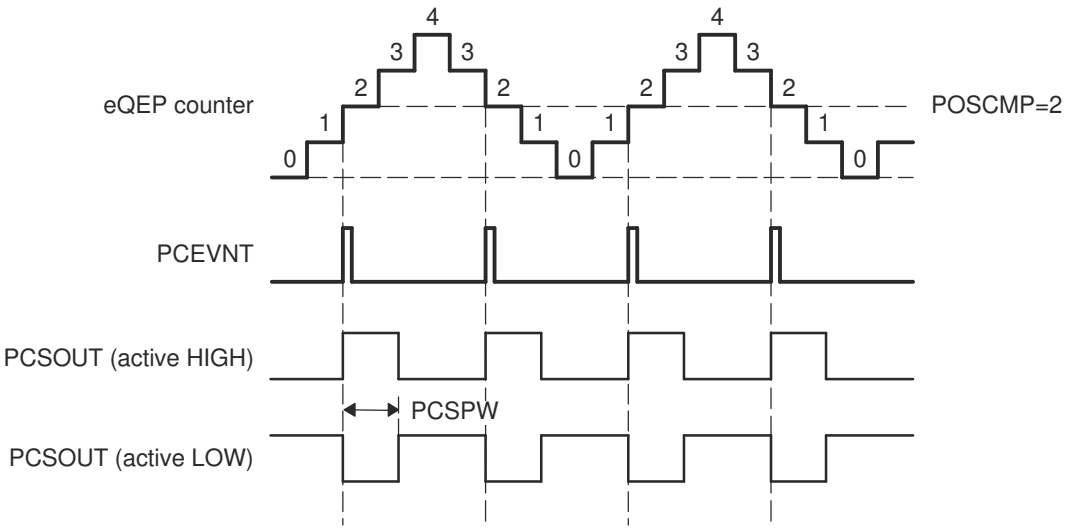


Figure 31-15. eQEP Position-compare Event Generation Points

The pulse stretcher logic in the position-compare unit generates a programmable position-compare sync pulse output on the position-compare match. In the event of a new position-compare match while a previous position-compare pulse is still active, then the pulse stretcher generates a pulse of specified duration from the new position-compare event as shown in Figure 31-16.

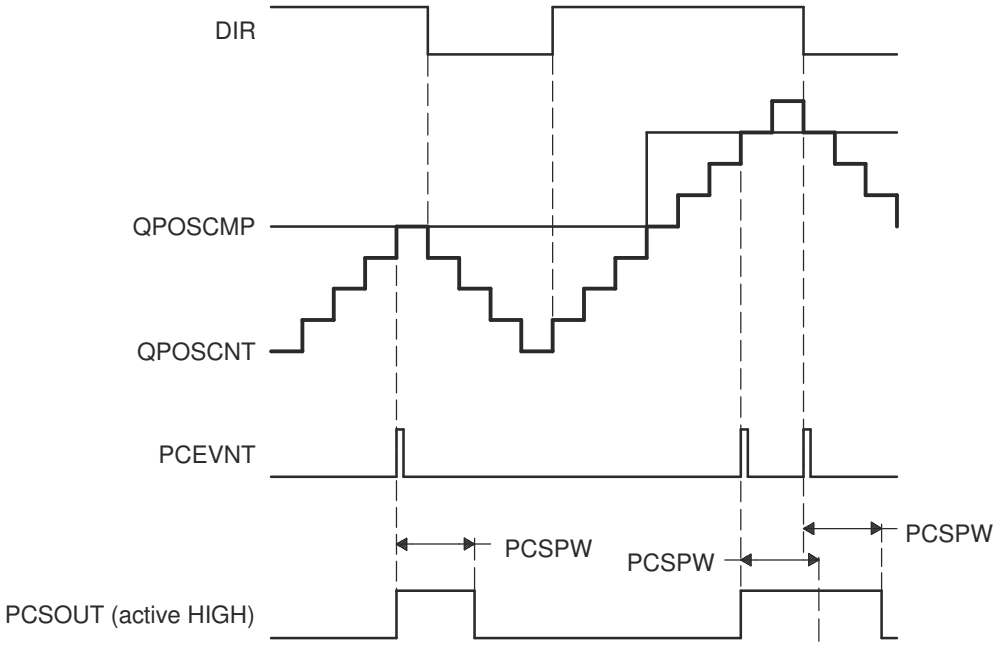


Figure 31-16. eQEP Position-compare Sync Output Pulse Stretcher

31.6 eQEP Edge Capture Unit

The eQEP peripheral includes an integrated edge capture unit to measure the elapsed time between the unit position events as shown in Figure 31-17. This feature is typically used for low-speed measurement using the following formula:

$$v(k) = \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \quad (26)$$

where:

- X = Unit position is defined by integer multiple of quadrature edges (see Figure 31-18)
- ΔT = Elapsed time between unit position events
- v(k) = Velocity at time instant "k"

The eQEP capture timer (QCTMR) runs from prescaled SYSCLKOUT and the prescaler is programmed by the QCAPCTL[CCPS] bits. The capture timer (QCTMR) value is latched into the capture period register (QCPRD) on every unit position event and then the capture timer is reset, a flag is set in QEPSTS:UPEVNT to indicate that new value is latched into the QCPRD register. Software can check this status flag before reading the period register for low speed measurement, and clear the flag by writing 1.

Time measurement (ΔT) between unit position events is correct if the following conditions are met:

- No more than 65,535 counts have occurred between unit position events.
- No direction change between unit position events.

If the QEP capture timer overflows between unit position events, then the timer sets the QEP capture overflow flag (QEPSTS[COEF]) in the status register and the QCPRDLAT register is set to 0xFFFF. If direction change occurs between the unit position events, then the error flag is set in the status register (QEPSTS[CDEF]) and the QCPRDLAT register is set to 0xFFFF.

The Capture Timer (QCTMR) and Capture Period register (QCPRD) can be configured to latch on the following events:

- CPU read of QPOSCNT register
- Unit time-out event

If the QEPCTL[QCLM] bit is cleared, then the capture timer and capture period values are latched into the QCTMRLAT and QCPRDLAT registers, respectively, when the CPU reads the position counter (QPOSCNT).

If the QEPCTL[QCLM] bit is set, then the position counter, capture timer, and capture period values are latched into the QPOSLAT, QCTMRLAT and QCPRDLAT registers, respectively, on unit time out.

Figure 31-19 shows the capture unit operation along with the position counter.

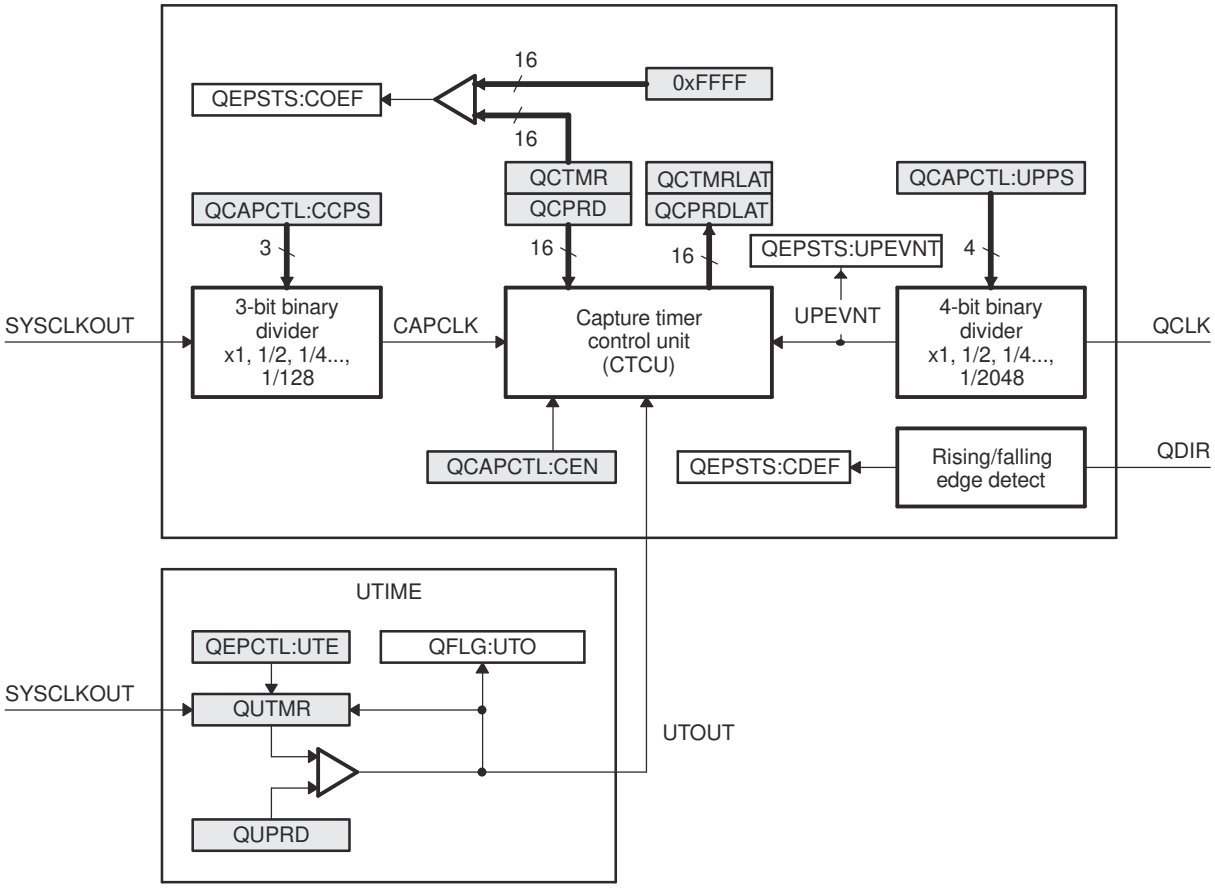
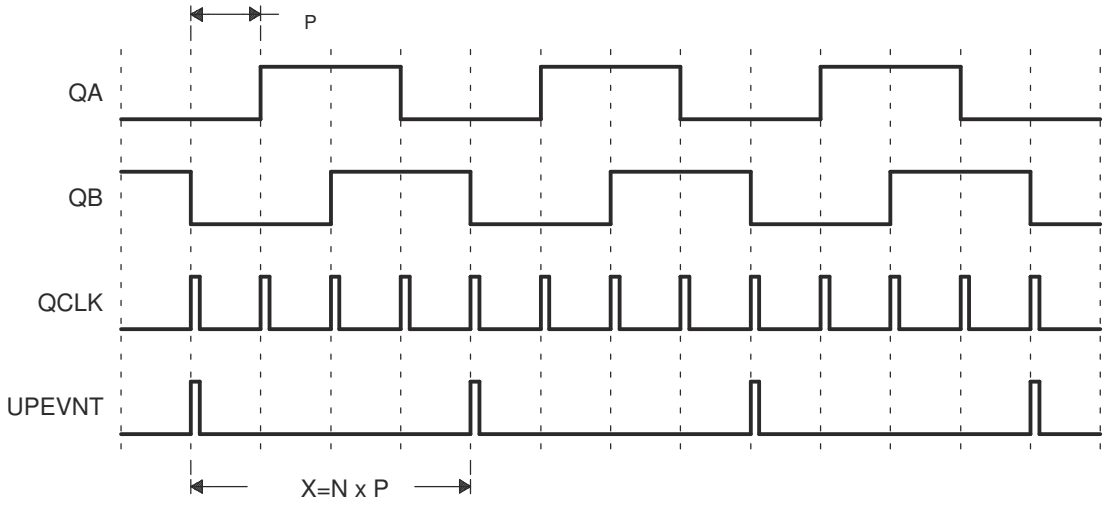


Figure 31-17. eQEP Edge Capture Unit

CAUTION

The QCAPCTL[UPPS] prescaler cannot be modified dynamically (such as switching the unit event prescaler from QCLK/4 to QCLK/8). Doing so can result in undefined behavior. The QCAPCTL[CPPS] prescaler can be modified dynamically (such as switching CAPCLK prescaling mode from SYSCLK/4 to SYSCLK/8) only after the capture unit is disabled.



N = Number of quadrature periods selected using QCAPCTL[UPPS] bits

Figure 31-18. Unit Position Event for Low Speed Measurement (QCAPCTL[UPPS] = 0010)

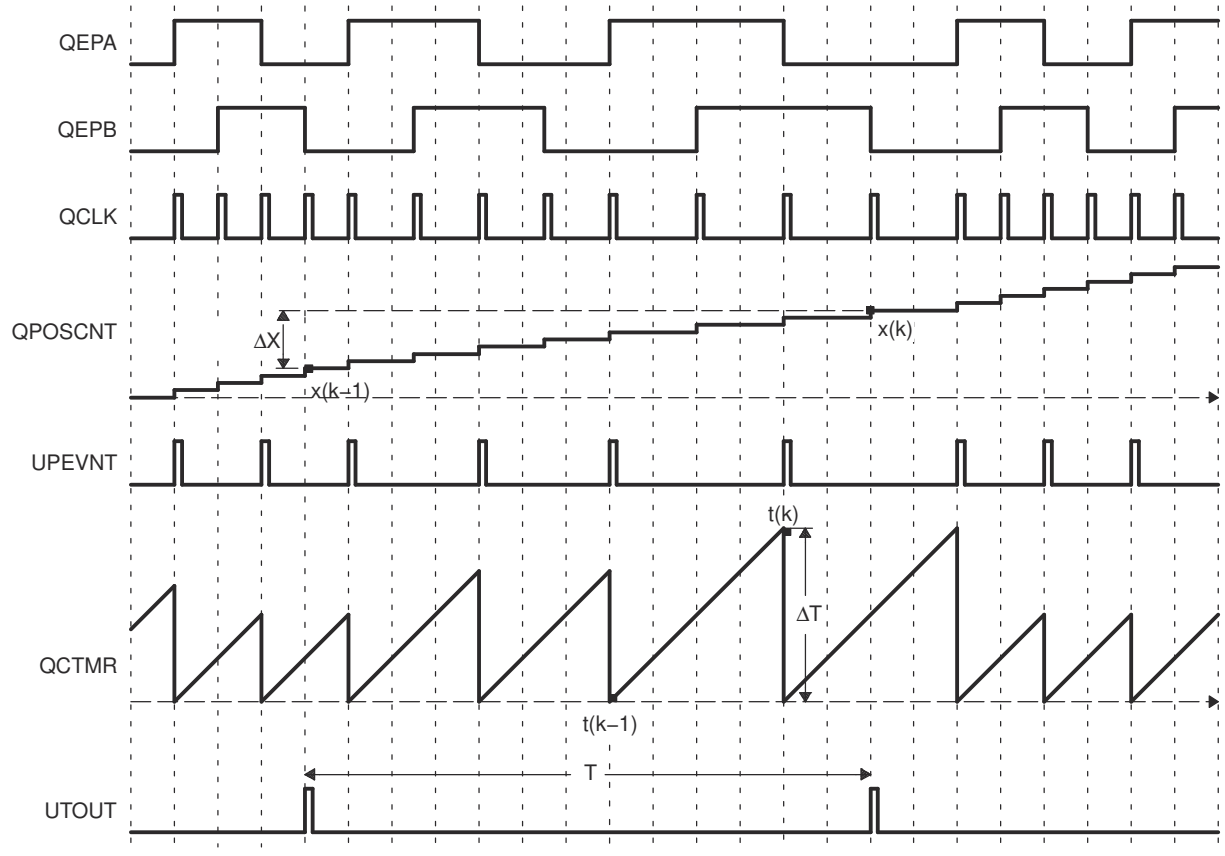


Figure 31-19. eQEP Edge Capture Unit - Timing Details

Velocity calculation equation:

$$v(k) = \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T} \text{ o} \quad (27)$$

where:

- v(k) = Velocity at time instant k
- x(k) = Position at time instant k
- x(k-1) = Position at time instant k-1
- T = Fixed unit time or inverse of velocity calculation rate
- ΔX = Incremental position movement in unit time
- X = Fixed unit position
- ΔT = Incremental time elapsed for unit position movement
- t(k) = Time instant "k"
- t(k-1) = Time instant "k-1"

Unit time (T) and unit period (X) are configured using the QUPRD and QCAPCTL[UPPS] registers. Incremental position output and incremental time output is available in the QPOSLAT and QCPRDLAT registers.

Parameter	Relevant Register to Configure or Read the Information
T	Unit Period Register (QUPRD)
ΔX	Incremental Position = QPOSLAT(k) - QPOSLAT(K-1)
X	Fixed-unit position defined by sensor resolution and QCAPCTL[UPPS] bits
ΔT	Capture Period Latch (QCPRDLAT)

31.7 eQEP Watchdog

The eQEP peripheral contains a 16-bit watchdog timer (Figure 31-20) that monitors the quadrature clock to indicate proper operation of the motion-control system. The eQEP watchdog timer is clocked from SYSCLKOUT/64 and the quadrature clock event (pulse) resets the watchdog timer. If no quadrature clock event is detected until a period match (QWDPRD = QWDTMR), then the watchdog timer times out and the watchdog interrupt flag is set (QFLG[WTO]). The time-out value is programmable through the watchdog period register (QWDPRD).

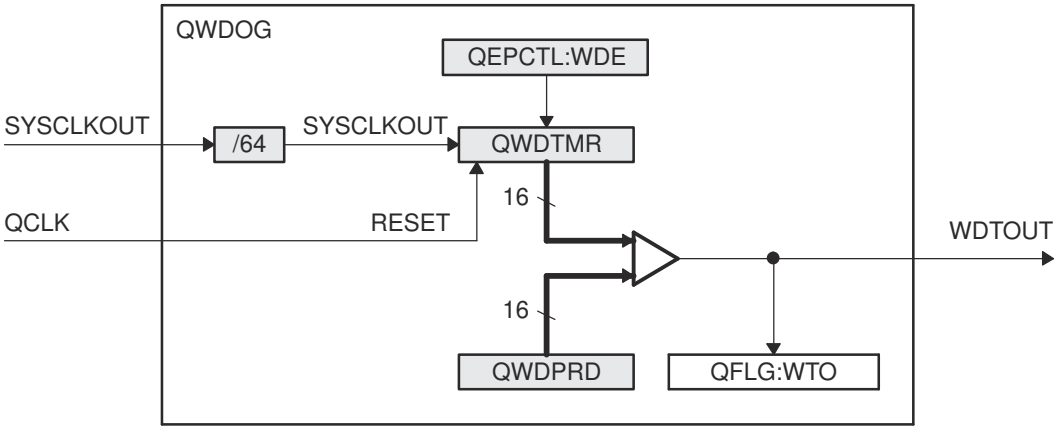


Figure 31-20. eQEP Watchdog Timer

31.8 eQEP Unit Timer Base

The eQEP peripheral includes a 32-bit timer (QUTMR) that is clocked by SYSCLKOUT to generate periodic interrupts for velocity calculations, see Figure 31-21. Whenever the unit timer (QUTMR) matches the unit period register (QUPRD), the eQEP peripheral resets the unit timer (QUTMR) and also generates the unit time out interrupt flag (QFLG[UTO]). The unit timer gets reset whenever timer value equals to configured period value.

The eQEP peripheral can be configured to latch the position counter, capture timer, and capture period values on a unit time out event so that latched values are used for velocity calculation as described in Section 31.6.

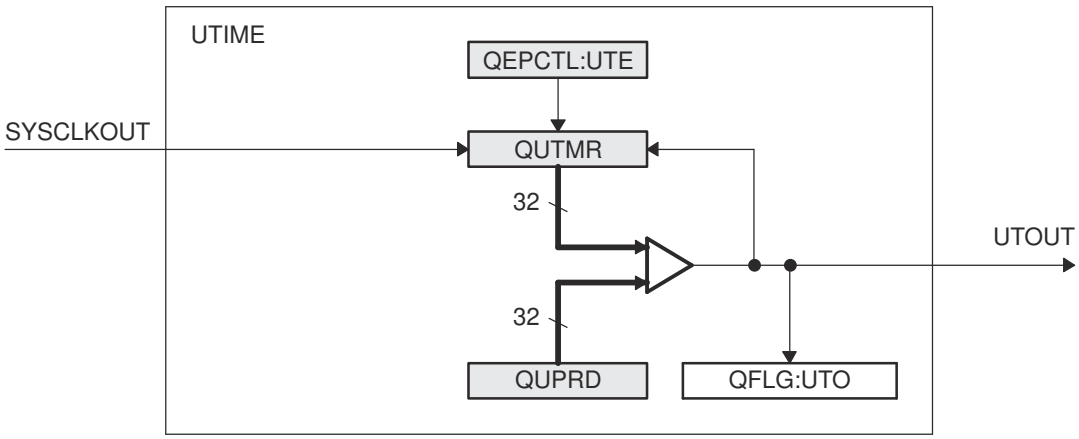


Figure 31-21. eQEP Unit Timer Base

31.9 QMA Module

The QEP Mode Adapter (QMA) is designed to extend the C2000™ eQEP module capabilities to support the additional modes described. Figure 31-22 depicts how the QMA module is integrated into to the eQEP module.

At reset, by default QMA logic is bypassed and the EQEPA and EQEPB inputs from the pins go directly into the eQEP module. When QMA module is enabled by configuring the QMACTRL[MODE] register, the EQEPA and EQEPB input are processed by this module and modified version of EQEPA and EQEPB signals are sent to the eQEP module. The QMA module requires the eQEP module to be configured in the Direction-Count mode and generates a clock signal on EQEPA input and direction signal on EQEPB input as needed for the proper operation of the intended mode.

- The xCLKMOD block inside the QMA module looks at the transitions on external EQEPA and EQEPB signals to generate the clock signal on the EQEPA input to the eQEP module.
- The xDIRMOD block inside the QMA module looks at the transitions on external EQEPA and EQEPB signals to generate the direction signal on the EQEPB input to the eQEP module.

The QMA module has error detection logic to detect illegal transitions on EQEPA and EQEPB input signals. The QMA module's error and interrupt are integrated inside the eQEP module as described in Section 31.10. In addition, the QMACTRL register configuration can be locked using the QMALOCK register. Refer to the register description for more details.

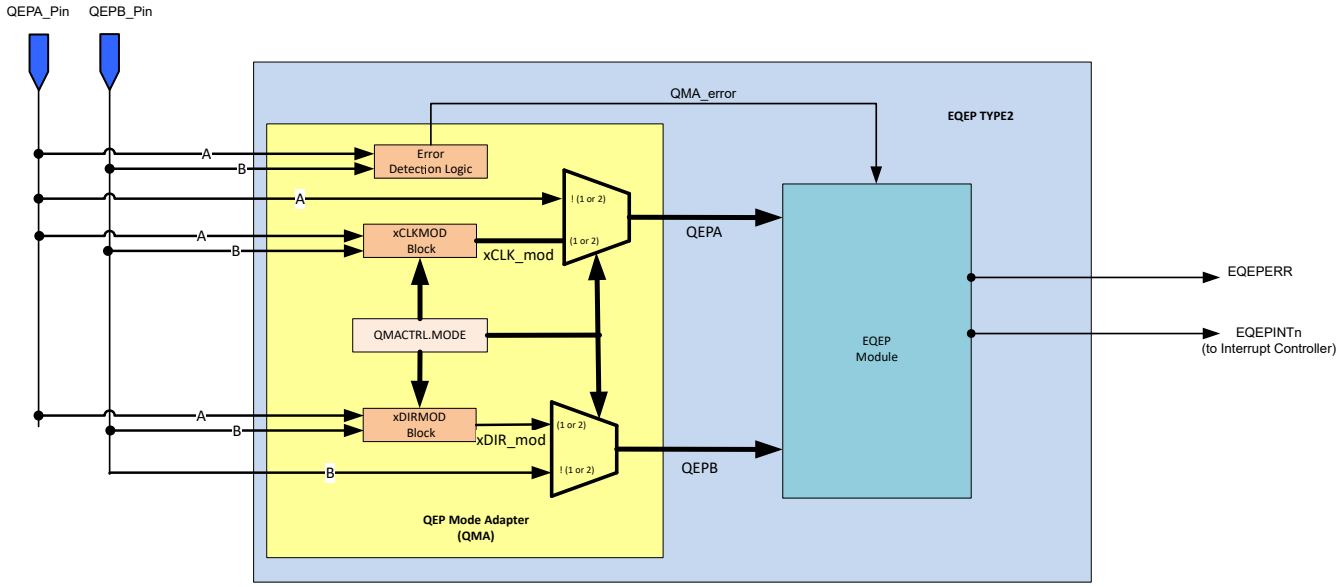


Figure 31-22. QMA Module Block Diagram

31.9.1 Modes of Operation

The QMA module can be operated in the following modes by configuring the QMACTRL register:

- QMA Mode-1 (QMACTRL[MODE] = 1)
- QMA Mode-2 (QMACTRL[MODE] = 2)

31.9.1.1 QMA Mode-1 (QMACTRL[MODE] = 1)

This mode is used when the default state of EQEPA and EQEPB inputs is high. In this mode, outputs of QMA correspond to the following as shown in Figure 31-23:

- EQEPA Output of QMA is the AND of EQEPA and EQEPB inputs coming from the pin
- EQEPB Output of QMA is the direction signal generated by QMA based on EQEPA and EQEPB inputs

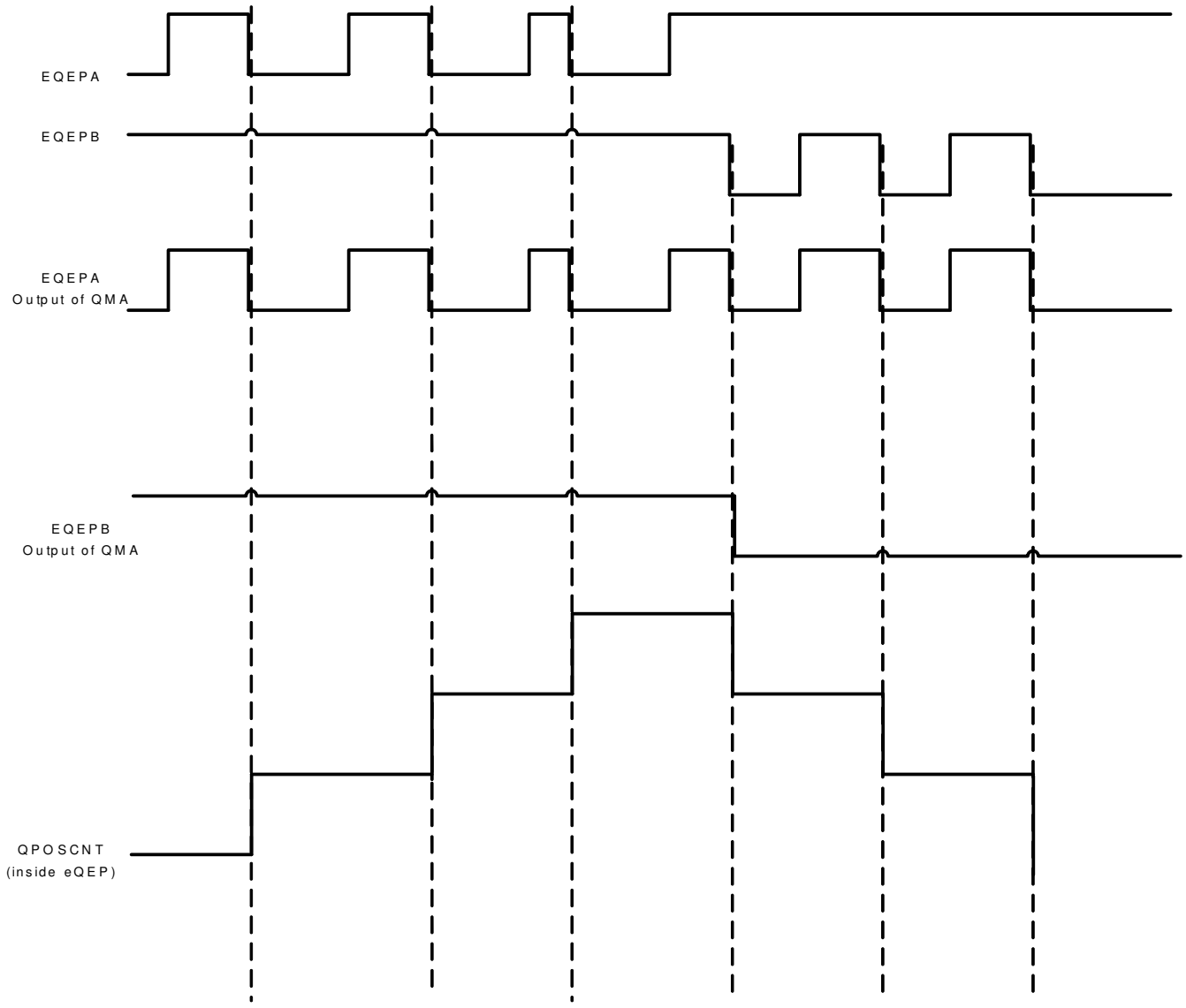


Figure 31-23. QMA Mode-1

31.9.1.2 QMA Mode-2 (QMACTRL[MODE] = 2)

This mode is used when the default state of EQEPA and EQEPB inputs is low. In this mode, outputs of QMA correspond to the following as shown in Figure 31-24:

- EQEPA Output of QMA is the OR of EQEPA and EQEPB inputs coming from the pin
- EQEPB Output of QMA is the direction signal generated by QMA based on EQEPA and EQEPB inputs

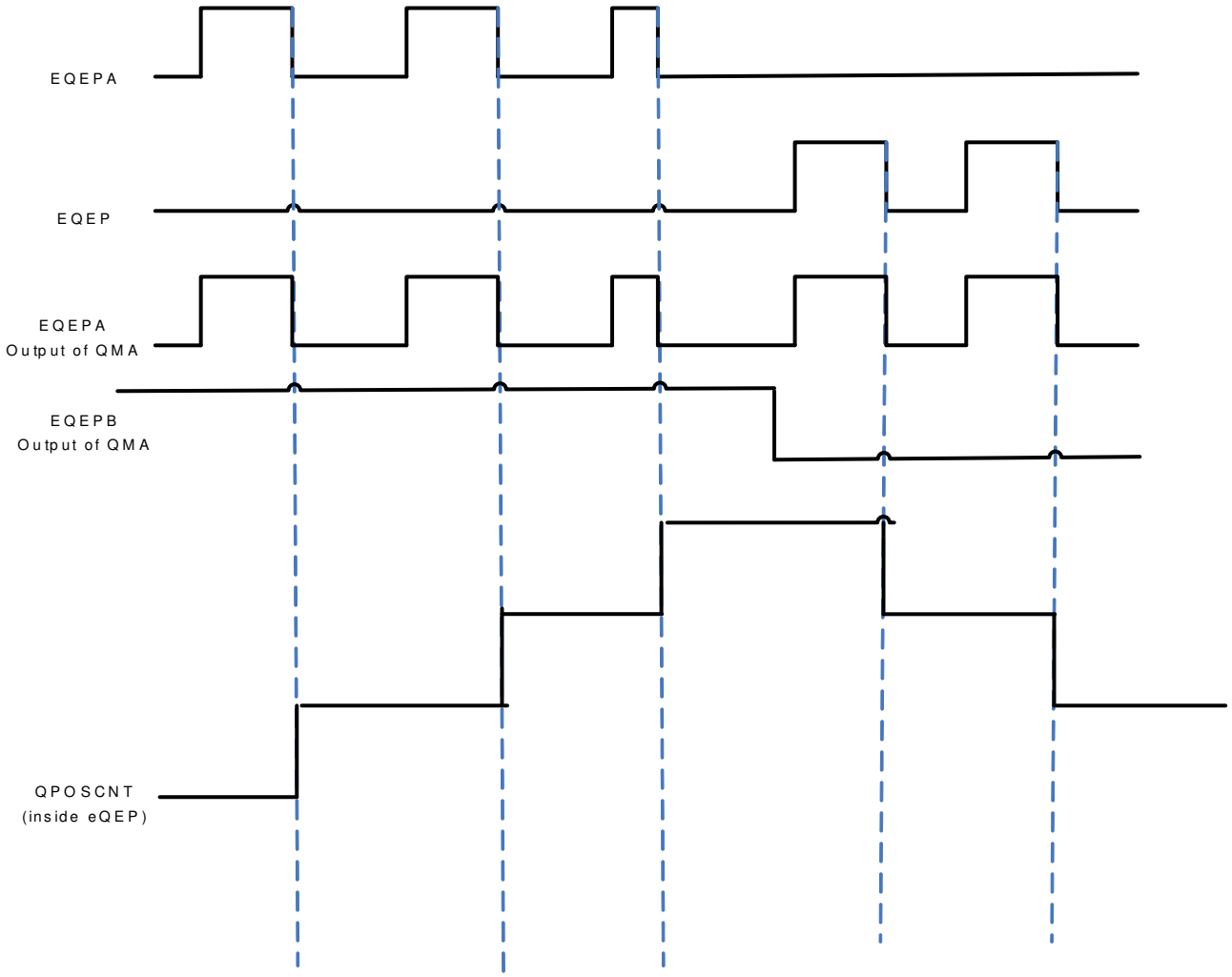


Figure 31-24. QMA Mode-2

31.9.2 Interrupt and Error Generation

The error detection logic detects illegal transitions on EQEPA and EQEPB signals and generates an error signal. This error signal can be used to generate eQEP interrupt and error output. Refer to Section 31.10 for details.

31.10 eQEP Interrupt Structure

Figure 31-25 shows how the interrupt mechanism works in the eQEP module.

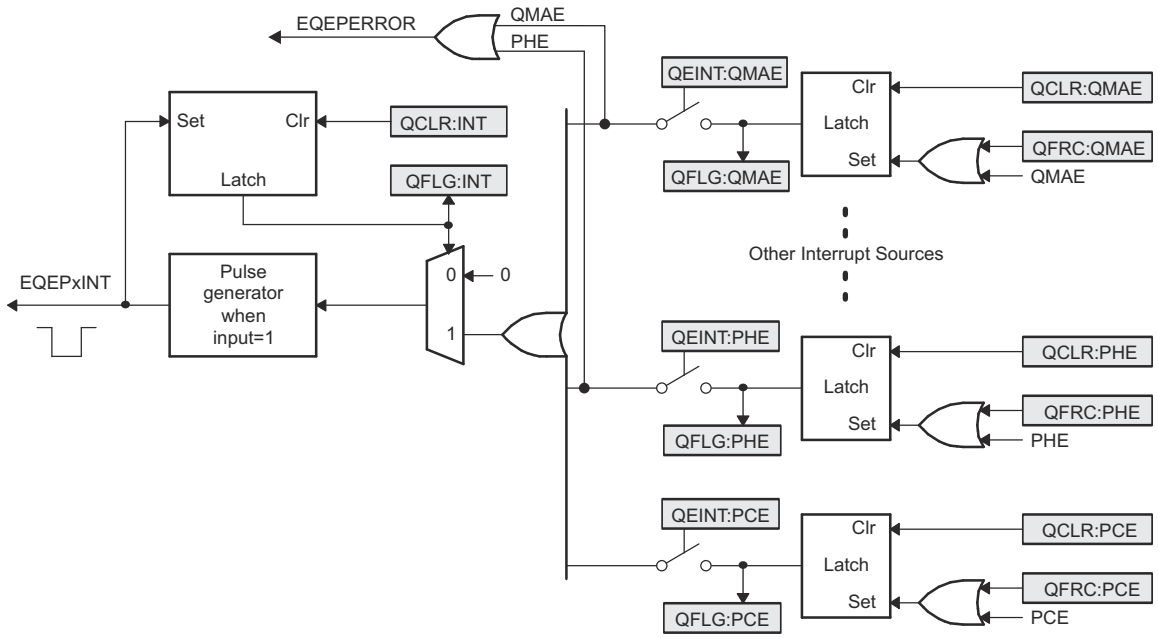


Figure 31-25. eQEP Interrupt Generation

Eleven interrupt events (PCE, PHE, QDC, WTO, PCU, PCO, PCR, PCM, SEL, IEL and UTO) can be generated. The interrupt control register (QEINT) is used to enable/disable individual interrupt event sources. The interrupt flag register (QFLG) indicates if any interrupt event has been latched and contains the global interrupt flag bit (INT).

An interrupt pulse is generated to PIPE when:

1. Interrupt is enabled for eQEP event inside QEINT register
2. Interrupt flag for eQEP event inside QFLG register is set, and
3. Global interrupt status flag bit QFLG[INT] had been cleared for previously generated interrupt event. The interrupt service routine needs to clear the global interrupt flag bit and the serviced event, by way of the interrupt clear register (QCLR), before any other interrupt pulses are generated. If either flags inside the QFLG register are not cleared, further interrupt events do not generate an interrupt to PIPE. You can force an interrupt event by way of the interrupt force register (QFRC), which is useful for test purposes.

31.11 Software

31.11.1 EQEP Registers to Driverlib Functions

Table 31-3. EQEP Registers to Driverlib Functions

File	Driverlib Function
QPOSCNT	
eqep.h	EQEP_getPosition
eqep.h	EQEP_setPosition
QPOSINIT	
eqep.h	EQEP_setInitialPosition
QPOSMAX	
eqep.h	EQEP_setPositionCounterConfig
QPOSCMP	
eqep.c	EQEP_setCompareConfig
QPOSILAT	
eqep.h	EQEP_getIndexPositionLatch
QPOSSLAT	
eqep.h	EQEP_getStrobePositionLatch
QPOSLAT	
eqep.h	EQEP_getPositionLatch
QUTMR	
-	
QUPRD	
eqep.h	EQEP_loadUnitTimer
eqep.h	EQEP_enableUnitTimer
QWDTMR	
eqep.h	EQEP_setWatchdogTimerValue
eqep.h	EQEP_getWatchdogTimerValue
QWDPRD	
eqep.h	EQEP_enableWatchdog
QDECCTL	
eqep.c	EQEP_setCompareConfig
eqep.c	EQEP_setInputPolarity
eqep.h	EQEP_setDecoderConfig
eqep.h	EQEP_enableDirectionChangeDuringIndex
eqep.h	EQEP_disableDirectionChangeDuringIndex
QEPCTL	
eqep.h	EQEP_enableModule
eqep.h	EQEP_disableModule
eqep.h	EQEP_setPositionCounterConfig
eqep.h	EQEP_enableUnitTimer
eqep.h	EQEP_disableUnitTimer
eqep.h	EQEP_enableWatchdog
eqep.h	EQEP_disableWatchdog
eqep.h	EQEP_setPositionInitMode
eqep.h	EQEP_setSWPositionInit
eqep.h	EQEP_setLatchMode

Table 31-3. EQEP Registers to Driverlib Functions (continued)

File	Driverlib Function
eqep.h	EQEP_setEmulationMode
QCAPCTL	
eqep.h	EQEP_setCaptureConfig
eqep.h	EQEP_enableCapture
eqep.h	EQEP_disableCapture
QPOSCTL	
eqep.c	EQEP_setCompareConfig
eqep.h	EQEP_enableCompare
eqep.h	EQEP_disableCompare
eqep.h	EQEP_setComparePulseWidth
QEINT	
eqep.h	EQEP_enableInterrupt
eqep.h	EQEP_disableInterrupt
QFLG	
eqep.h	EQEP_getInterruptStatus
eqep.h	EQEP_getError
QCLR	
eqep.h	EQEP_clearInterruptStatus
QFRC	
eqep.h	EQEP_forceInterrupt
QEPSTS	
eqep.h	EQEP_getDirection
eqep.h	EQEP_getStatus
eqep.h	EQEP_clearStatus
QCTMR	
eqep.h	EQEP_getCaptureTimer
eqep.h	EQEP_getCaptureTimerLatch
QCPRD	
eqep.h	EQEP_getCapturePeriod
eqep.h	EQEP_getCapturePeriodLatch
QCTMRLAT	
eqep.h	EQEP_getCaptureTimerLatch
QCPRDLAT	
eqep.h	EQEP_getCapturePeriodLatch
REV	
-	
QEPSTROBESEL	
eqep.h	EQEP_setStrobeSource
QMACTRL	
eqep.h	EQEP_setQMAModuleMode
QEPSRCSEL	
eqep.h	EQEP_selectSource

31.11.2 EQEP Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/eqep

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

31.11.2.1 Frequency Measurement Using eQEP via unit timeout interrupt - SINGLE_CORE

FILE: eqep_ex2_freq_cal_interrupt.c

This example will calculate the frequency of an input signal using the eQEP module. ePWM1A is configured to generate this input signal with a frequency of 5 kHz. EQEP unit timeout is set which will generate an interrupt every *UNIT_PERIOD* microseconds and frequency calculation occurs continuously

The configuration for this example is as follows
- PWM frequency is specified as 5000Hz
- UNIT_PERIOD is specified as 10000 us
- Min frequency is (1/(2*10ms)) i.e 50Hz
- Highest frequency can be (2^32)/ ((2*10ms))
- Resolution of frequency measurement is 50hz

freq : Frequency Measurement is obtained by counting the external input pulses for UNIT_PERIOD (unit timer set to 10 ms).

External Connections
- Connect GPIO20/eQEP1A to GPIO0/ePWM1A

Watch Variables
- *freq* - Frequency measurement using position counter/unit time out
- *pass* - If measured frequency matches with PWM frequency then pass = 1 else 0

31.11.2.2 Motor speed and direction measurement using eQEP via unit timeout interrupt - SINGLE_CORE

FILE: eqep_ex5_speed_dir_motor.c

This example can be used to sense the speed and direction of motor using eQEP in quadrature encoder mode. ePWM1A is configured to simulate motor encoder signals with frequency of 5 kHz on both A and B pins with 90 degree phase shift (so as to run this example without motor). EQEP unit timeout is set which will generate an interrupt every *UNIT_PERIOD* microseconds and speed calculation occurs continuously based on the direction of motor

The configuration for this example is as follows
- PWM frequency is specified as 5000Hz
- UNIT_PERIOD is specified as 10000 us
- Simulated quadrature signal frequency is 20000Hz (4 * 5000)
- Encoder holes assumed as 1000
- Thus Simulated motor speed is 300rpm (5000 * (60 / 1000))

freq : Simulated quadrature signal frequency measured by counting the external input pulses for UNIT_PERIOD (unit timer set to 10 ms). *speed* : Measure motor speed in rpm *dir* : Indicates clockwise (1) or anticlockwise (-1)

External Connections (if motor encoder signals are simulated by ePWM)
- Connect GPIO20/eQEP1A to GPIO0/ePWM1A
- Connect GPIO21/eQEP1B to GPIO1/ePWM1B With motor
- Comment in "MOTOR" in includes
- Connect GPIO20/eQEP1A to encoder A output
- Connect GPIO21/eQEP1B to encoder B output

Watch Variables

- *freq* : Simulated motor frequency measurement is obtained by counting the external input pulses for UNIT_PERIOD (unit timer set to 10 ms).
- *speed* : Measure motor speed in rpm
- *dir* : Indicates clockwise (1) or anticlockwise (-1)
- *pass* - If measured qudrature frequency matches with i.e. input quadrature frequency (4 * PWM frequency) then pass = 1 else fail = 1 (** only when "MOTOR" is commented out)

31.12 EQEP Registers

This Section describes the EQEP Registers.

31.12.1 EQEP Base Address Table

Table 31-4. EQEP Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
EQEP_REGS	EQEP1_BASE	0x7008_8000	YES	YES	YES	YES	YES	YES	-	YES
EQEP_REGS	EQEP2_BASE	0x7008_9000	YES	YES	YES	YES	YES	YES	-	YES
EQEP_REGS	EQEP3_BASE	0x7008_A000	YES	YES	YES	YES	YES	YES	-	YES
EQEP_REGS	EQEP4_BASE	0x7008_B000	YES	YES	YES	YES	YES	YES	-	YES
EQEP_REGS	EQEP5_BASE	0x7008_C000	YES	YES	YES	YES	YES	YES	-	YES
EQEP_REGS	EQEP6_BASE	0x7008_D000	YES	YES	YES	YES	YES	YES	-	YES

31.12.2 EQEP_REGS Registers

Table 31-5 lists the memory-mapped registers for the EQEP_REGS registers. All register offset addresses not listed in Table 31-5 should be considered as reserved locations and the register contents should not be modified.

Table 31-5. EQEP_REGS Registers

Offset	Acronym	Register Name	Protection
0h	QPOSCNT	Position Counter	
4h	QPOSINIT	Position Counter Init	
8h	QPOSMAX	Maximum Position Count	
Ch	QPOSCMP	Position Compare	
10h	QPOSILAT	Index Position Latch	
14h	QPOSSLAT	Strobe Position Latch	
18h	QPOSLAT	Position Latch	
1Ch	QUTMR	QEP Unit Timer	
20h	QUPRD	QEP Unit Period	
24h	QWDTMR	QEP Watchdog Timer	
26h	QWDPRD	QEP Watchdog Period	
28h	QDECCTL	Quadrature Decoder Control	
2Ah	QEPCTL	QEP Control	
2Ch	QCAPCTL	Qaudrature Capture Control	
2Eh	QPOSCTL	Position Compare Control	
30h	QEINT	QEP Interrupt Control	
32h	QFLG	QEP Interrupt Flag	
34h	QCLR	QEP Interrupt Clear	
36h	QFRC	QEP Interrupt Force	
38h	QEPSTS	QEP Status	
3Ah	QCTMR	QEP Capture Timer	
3Ch	QCPRD	QEP Capture Period	
3Eh	QCTMRLAT	QEP Capture Latch	
40h	QCPRDLAT	QEP Capture Period Latch	
60h	REV	QEP Revision Number	
64h	QEPSTROBESEL	QEP Strobe select register	
68h	QMACTRL	QMA Control register	
6Ch	QEPSRCSEL	QEP Source Select Register	

Complex bit access types are encoded to fit into small table cells. Table 31-6 shows the codes that are used for access types in this section.

Table 31-6. EQEP_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear

Table 31-6. EQEP_REGS Access Type Codes (continued)

Access Type	Code	Description
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

31.12.2.1 QPOSCNT Register (Offset = 0h) [Reset = 00000000h]

QPOSCNT is shown in Figure 31-26 and described in Table 31-7.

Return to the Summary Table.

Position Counter

Figure 31-26. QPOSCNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSCNT																															
R/W-0h																															

Table 31-7. QPOSCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSCNT	R/W	0h	Position Counter This 32-bit position counter register counts up/down on every eQEP pulse based on direction input. This counter acts as a position integrator whose count value is proportional to position from a give reference point. This Register acts as a Read ONLY register while counter is counting up/down. Note: It is recommended to only write to the position counter register (QPOSCNT) during initialization, i.e. when the eQEP position counter is disabled (QPEN bit of QEPCTL is zero). Once the position counter is enabled (QPEN bit is one), writing to the eQEP position counter register (QPOSCNT) may cause unexpected results. Reset type: SYSRSn

31.12.2.2 QPOSINIT Register (Offset = 4h) [Reset = 00000000h]

QPOSINIT is shown in Figure 31-27 and described in Table 31-8.

Return to the Summary Table.

Position Counter Init

Figure 31-27. QPOSINIT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSINIT																															
R/W-0h																															

Table 31-8. QPOSINIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSINIT	R/W	0h	Position Counter Init This register contains the position value that is used to initialize the position counter based on external strobe or index event. The position counter can be initialized through software. Writes to this register should always be full 32-bit writes. Reset type: SYSRSn

31.12.2.3 QPOSMAX Register (Offset = 8h) [Reset = 00000000h]

QPOSMAX is shown in Figure 31-28 and described in Table 31-9.

Return to the Summary Table.

Maximum Position Count

Figure 31-28. QPOSMAX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSMAX																															
R/W-0h																															

Table 31-9. QPOSMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSMAX	R/W	0h	Maximum Position Count This register contains the maximum position counter value. Writes to this register should always be full 32-bit writes. Reset type: SYSRSn

31.12.2.4 QPOSCMP Register (Offset = Ch) [Reset = 00000000h]

QPOSCMP is shown in Figure 31-29 and described in Table 31-10.

Return to the Summary Table.

Position Compare

Figure 31-29. QPOSCMP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSCMP																															
R/W-0h																															

Table 31-10. QPOSCMP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSCMP	R/W	0h	Position Compare The position-compare value in this register is compared with the position counter (QPOSCNT) to generate sync output and/or interrupt on compare match. Writes to this register should always be full 32-bit writes. Reset type: SYSRSn

31.12.2.5 QPOSILAT Register (Offset = 10h) [Reset = 00000000h]

QPOSILAT is shown in Figure 31-30 and described in Table 31-11.

Return to the Summary Table.

Index Position Latch

Figure 31-30. QPOSILAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSILAT																															
R-0h																															

Table 31-11. QPOSILAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSILAT	R	0h	Index Position Latch The position-counter value is latched into this register on an index event as defined by the QEPCTL[IEL] bits. Reset type: SYSRSn

31.12.2.6 QPOSSLAT Register (Offset = 14h) [Reset = 00000000h]

QPOSSLAT is shown in Figure 31-31 and described in Table 31-12.

Return to the Summary Table.

Strobe Position Latch

Figure 31-31. QPOSSLAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSSLAT																															
R-0h																															

Table 31-12. QPOSSLAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSSLAT	R	0h	Strobe Position Latch The position-counter value is latched into this register on a strobe event as defined by the QEPCTL[SEL] bits. Reset type: SYSRSn

31.12.2.7 QPOSLAT Register (Offset = 18h) [Reset = 00000000h]

QPOSLAT is shown in Figure 31-32 and described in Table 31-13.

Return to the Summary Table.

Position Latch

Figure 31-32. QPOSLAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSLAT																															
R-0h																															

Table 31-13. QPOSLAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSLAT	R	0h	Position Latch The position-counter value is latched into this register on a unit time out event. Reset type: SYSRSn

31.12.2.8 QUTMR Register (Offset = 1Ch) [Reset = 00000000h]

QUTMR is shown in Figure 31-33 and described in Table 31-14.

Return to the Summary Table.

QEP Unit Timer

Figure 31-33. QUTMR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUTMR																															
R/W-0h																															

Table 31-14. QUTMR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QUTMR	R/W	0h	QEP Unit Timer This register acts as time base for unit time event generation. When this timer value matches the unit time period value a unit time event is generated. Writes to this register should always be full 32-bit writes. Reset type: SYSRSn

31.12.2.9 QUPRD Register (Offset = 20h) [Reset = 00000000h]

QUPRD is shown in Figure 31-34 and described in Table 31-15.

Return to the Summary Table.

QEP Unit Period

Figure 31-34. QUPRD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUPRD																															
R/W-0h																															

Table 31-15. QUPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QUPRD	R/W	0h	QEP Unit Period This register contains the period count for the unit timer to generate periodic unit time events. These events latch the eQEP position information at periodic intervals and optionally generate an interrupt. Writes to this register should always be full 32-bit writes. Reset type: SYSRSn

31.12.2.10 QWDTMR Register (Offset = 24h) [Reset = 0000h]

QWDTMR is shown in Figure 31-35 and described in Table 31-16.

Return to the Summary Table.

QEP Watchdog Timer

Figure 31-35. QWDTMR Register

15	14	13	12	11	10	9	8
QWDTMR							
R/W-0h							
7	6	5	4	3	2	1	0
QWDTMR							
R/W-0h							

Table 31-16. QWDTMR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QWDTMR	R/W	0h	QEP Watchdog Timer This register acts as time base for the watchdog to detect motor stalls. When this timer value matches with the watchdog's period value a watchdog timeout interrupt is generated. This register is reset upon edge transition in quadrature-clock indicating the motion. Reset type: SYSRSn

31.12.2.11 QWDPRD Register (Offset = 26h) [Reset = 0000h]

QWDPRD is shown in Figure 31-36 and described in Table 31-17.

Return to the Summary Table.

QEP Watchdog Period

Figure 31-36. QWDPRD Register

15	14	13	12	11	10	9	8
QWDPRD							
R/W-0h							
7	6	5	4	3	2	1	0
QWDPRD							
R/W-0h							

Table 31-17. QWDPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QWDPRD	R/W	0h	QEP Watchdog Period This register contains the time-out count for the eQEP peripheral watch dog timer. When the watchdog timer value matches the watchdog period value, a watchdog timeout interrupt is generated. Reset type: SYSRSn

31.12.2.12 QDECCTL Register (Offset = 28h) [Reset = 0000h]

QDECCTL is shown in Figure 31-37 and described in Table 31-18.

Return to the Summary Table.

Quadrature Decoder Control

Figure 31-37. QDECCTL Register

15	14	13	12	11	10	9	8
QSRC		SOEN	SPSEL	XCR	SWAP	IGATE	QAP
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
QBP	QIP	QSP	RESERVED				QIDIRE
R/W-0h	R/W-0h	R/W-0h	R-0h				R/W-0h

Table 31-18. QDECCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	QSRC	R/W	0h	Position-counter source selection Reset type: SYSRSn 0h (R/W) = Quadrature count mode (QCLK = iCLK, QDIR = iDIR) 1h (R/W) = Direction-count mode (QCLK = xCLK, QDIR = xDIR) 2h (R/W) = UP count mode for frequency measurement (QCLK = xCLK, QDIR = 1) 3h (R/W) = DOWN count mode for frequency measurement (QCLK = xCLK, QDIR = 0)
13	SOEN	R/W	0h	Sync output-enable Reset type: SYSRSn 0h (R/W) = Disable position-compare sync output 1h (R/W) = Enable position-compare sync output
12	SPSEL	R/W	0h	Sync output pin selection Reset type: SYSRSn 0h (R/W) = Index pin is used for sync output 1h (R/W) = Strobe pin is used for sync output
11	XCR	R/W	0h	External Clock Rate Reset type: SYSRSn 0h (R/W) = 2x resolution: Count the rising/falling edge 1h (R/W) = 1x resolution: Count the rising edge only
10	SWAP	R/W	0h	CLK/DIR Signal Source for Position Counter Reset type: SYSRSn 0h (R/W) = Quadrature-clock inputs are not swapped 1h (R/W) = Quadrature-clock inputs are swapped
9	IGATE	R/W	0h	Index pulse gating option Reset type: SYSRSn 0h (R/W) = Disable gating of Index pulse 1h (R/W) = Gate the index pin with strobe
8	QAP	R/W	0h	QEPA input polarity Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Negates QEPA input
7	QBP	R/W	0h	QEPB input polarity Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Negates QEPB input
6	QIP	R/W	0h	QEPI input polarity Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Negates QEPI input

Table 31-18. QDECCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	QSP	R/W	0h	QEPS input polarity Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Negates QEPS input
4-1	RESERVED	R	0h	Reserved
0	QIDIRE	R/W	0h	0 - Compatible mode, Behavior same as existing devices 1 - Enhancement for Direction change during Index will be enabled: On QEPI direction change, the incoming posedge of QA can erroneously update/reset the position counter of the eQEP. This bit only needs to be enabled if the application requires a direction change occurring at the same time as an incoming QEPI signal, or when erroneous PC resets are observed. Reset type: SYSRSn

31.12.2.13 QEPCTL Register (Offset = 2Ah) [Reset = 0000h]

QEPCTL is shown in Figure 31-38 and described in Table 31-19.

Return to the Summary Table.

QEP Control

Figure 31-38. QEPCTL Register

15	14	13	12	11	10	9	8
FREE_SOFT		PCRM		SEI		IEI	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
SWI	SEL	IEL		QPEN	QCLM	UTE	WDE
R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 31-19. QEPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	FREE_SOFT	R/W	0h	Emulation mode Reset type: SYSRSn 0h (R/W) = QPOSCNT behavior Position counter stops immediately on emulation suspend 0h (R/W) = QWDTMR behavior Watchdog counter stops immediately 0h (R/W) = QUTMR behavior Unit timer stops immediately 0h (R/W) = QCTMR behavior Capture Timer stops immediately 1h (R/W) = QPOSCNT behavior Position counter continues to count until the rollover 1h (R/W) = QWDTMR behavior Watchdog counter counts until WD period match roll over 1h (R/W) = QUTMR behavior Unit timer counts until period rollover 1h (R/W) = QCTMR behavior Capture Timer counts until next unit period event 2h (R/W) = QPOSCNT behavior Position counter is unaffected by emulation suspend 2h (R/W) = QWDTMR behavior Watchdog counter is unaffected by emulation suspend 2h (R/W) = QUTMR behavior Unit timer is unaffected by emulation suspend 2h (R/W) = QCTMR behavior Capture Timer is unaffected by emulation suspend 3h (R/W) = Same as FREE_SOFT_2
13-12	PCRM	R/W	0h	Postion counter reset Reset type: SYSRSn 0h (R/W) = Position counter reset on an index event 1h (R/W) = Position counter reset on the maximum position 2h (R/W) = Position counter reset on the first index event 3h (R/W) = Position counter reset on a unit time event
11-10	SEI	R/W	0h	Strobe event initialization of position counter Reset type: SYSRSn 0h (R/W) = Does nothing (action disabled) 1h (R/W) = Does nothing (action disabled) 2h (R/W) = Initializes the position counter on rising edge of the QEPS signal 3h (R/W) = Clockwise Direction: Initializes the position counter on the rising edge of QEPS strobe Counter Clockwise Direction: Initializes the position counter on the falling edge of QEPS strobe

Table 31-19. QEPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9-8	IEI	R/W	0h	Index event init of position count Reset type: SYSRSn 0h (R/W) = Do nothing (action disabled) 1h (R/W) = Do nothing (action disabled) 2h (R/W) = Initializes the position counter on the rising edge of the QEPI signal (QPOSCNT = QPOSINIT) 3h (R/W) = Initializes the position counter on the falling edge of QEPI signal (QPOSCNT = QPOSINIT)
7	SWI	R/W	0h	Software init position counter Reset type: SYSRSn 0h (R/W) = Do nothing (action disabled) 1h (R/W) = Initialize position counter (QPOSCNT=QPOSINIT). This bit is not cleared automatically
6	SEL	R/W	0h	Strobe event latch of position counter Reset type: SYSRSn 0h (R/W) = The position counter is latched on the rising edge of QEPS strobe (QPOSSLAT = POSCCNT). Latching on the falling edge can be done by inverting the strobe input using the QSP bit in the QDECCTL register 1h (R/W) = Clockwise Direction: Position counter is latched on rising edge of QEPS strobe Counter Clockwise Direction: Position counter is latched on falling edge of QEPS strobe
5-4	IEL	R/W	0h	Index event latch of position counter (software index marker) Reset type: SYSRSn 0h (R/W) = Reserved 1h (R/W) = Latches position counter on rising edge of the index signal 2h (R/W) = Latches position counter on falling edge of the index signal 3h (R/W) = Software index marker. Latches the position counter and quadrature direction flag on index event marker. The position counter is latched to the QPOSILAT register and the direction flag is latched in the QEPSTS[QDLF] bit. This mode is useful for software index marking.
3	QPEN	R/W	0h	Quadrature position counter enable/software reset Reset type: SYSRSn 0h (R/W) = Reset the eQEP peripheral internal operating flags/read-only registers. Control/configuration registers are not disturbed by a software reset. When QPEN is disabled, some flags in the QFLG register do not get reset or cleared and show the actual state of that flag. 1h (R/W) = eQEP position counter is enabled
2	QCLM	R/W	0h	QEP capture latch mode Reset type: SYSRSn 0h (R/W) = Latch on position counter read by CPU. Capture timer and capture period values are latched into QCTMRLAT and QCPRDLAT registers when CPU reads the QPOSCNT register. 1h (R/W) = Latch on unit time out. Position counter, capture timer and capture period values are latched into QPOSLAT, QCTMRLAT and QCPRDLAT registers on unit time out.
1	UTE	R/W	0h	QEP unit timer enable Reset type: SYSRSn 0h (R/W) = Disable eQEP unit timer 1h (R/W) = Enable unit timer
0	WDE	R/W	0h	QEP watchdog enable Reset type: SYSRSn 0h (R/W) = Disable the eQEP watchdog timer 1h (R/W) = Enable the eQEP watchdog timer

31.12.2.14 QCAPCTL Register (Offset = 2Ch) [Reset = 0000h]

QCAPCTL is shown in Figure 31-39 and described in Table 31-20.

Return to the Summary Table.

Qaudrature Capture Control

Figure 31-39. QCAPCTL Register

15	14	13	12	11	10	9	8
CEN	RESERVED						
R/W-0h	R-0h						
7	6	5	4	3	2	1	0
RESERVED	CCPS			UPPS			
R-0h	R/W-0h			R/W-0h			

Table 31-20. QCAPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	CEN	R/W	0h	Enable eQEP capture Reset type: SYSRSn 0h (R/W) = eQEP capture unit is disabled 1h (R/W) = eQEP capture unit is enabled
14-7	RESERVED	R	0h	Reserved
6-4	CCPS	R/W	0h	eQEP capture timer clock prescaler Reset type: SYSRSn 0h (R/W) = CAPCLK = SYSCLKOUT/1 1h (R/W) = CAPCLK = SYSCLKOUT/2 2h (R/W) = CAPCLK = SYSCLKOUT/4 3h (R/W) = CAPCLK = SYSCLKOUT/8 4h (R/W) = CAPCLK = SYSCLKOUT/16 5h (R/W) = CAPCLK = SYSCLKOUT/32 6h (R/W) = CAPCLK = SYSCLKOUT/64 7h (R/W) = CAPCLK = SYSCLKOUT/128
3-0	UPPS	R/W	0h	Unit position event prescaler Reset type: SYSRSn 0h (R/W) = UPEVNT = QCLK/1 1h (R/W) = UPEVNT = QCLK/2 2h (R/W) = UPEVNT = QCLK/4 3h (R/W) = UPEVNT = QCLK/8 4h (R/W) = UPEVNT = QCLK/16 5h (R/W) = UPEVNT = QCLK/32 6h (R/W) = UPEVNT = QCLK/64 7h (R/W) = UPEVNT = QCLK/128 8h (R/W) = UPEVNT = QCLK/256 9h (R/W) = UPEVNT = QCLK/512 Ah (R/W) = UPEVNT = QCLK/1024 Bh (R/W) = UPEVNT = QCLK/2048 Ch (R/W) = Reserved Dh (R/W) = Reserved Eh (R/W) = Reserved Fh (R/W) = Reserved

31.12.2.15 QPOSCTL Register (Offset = 2Eh) [Reset = 0000h]

QPOSCTL is shown in Figure 31-40 and described in Table 31-21.

Return to the Summary Table.

Position Compare Control

Figure 31-40. QPOSCTL Register

15	14	13	12	11	10	9	8
PCSHDW	PCLOAD	PCPOL	PCE	PCSPW			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h			
7	6	5	4	3	2	1	0
PCSPW							
R/W-0h							

Table 31-21. QPOSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	PCSHDW	R/W	0h	Position compare of shadow enable Reset type: SYSRSn 0h (R/W) = Shadow disabled, load Immediate 1h (R/W) = Shadow enabled
14	PCLOAD	R/W	0h	Position compare of shadow load Reset type: SYSRSn 0h (R/W) = Load on QPOSCNT = 0 1h (R/W) = Load when QPOSCNT = QPOSCMP
13	PCPOL	R/W	0h	Polarity of sync output Reset type: SYSRSn 0h (R/W) = Active HIGH pulse output 1h (R/W) = Active LOW pulse output
12	PCE	R/W	0h	Position compare enable/disable Reset type: SYSRSn 0h (R/W) = Disable position compare unit 1h (R/W) = Enable position compare unit
11-0	PCSPW	R/W	0h	Select-position-compare sync output pulse width Reset type: SYSRSn 0h (R/W) = 1 * 4 * SYSCLKOUT cycles 1h (R/W) = 2 * 4 * SYSCLKOUT cycles FFFh (R/W) = 4096 * 4 * SYSCLKOUT cycles

31.12.2.16 QEINT Register (Offset = 30h) [Reset = 0000h]

QEINT is shown in Figure 31-41 and described in Table 31-22.

Return to the Summary Table.

QEP Interrupt Control

Figure 31-41. QEINT Register

15	14	13	12	11	10	9	8
RESERVED			QMAE	UTO	IEL	SEL	PCM
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	QPE	PCE	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 31-22. QEINT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	QMAE	R/W	0h	QMA Error Interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
11	UTO	R/W	0h	Unit time out interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
10	IEL	R/W	0h	Index event latch interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
9	SEL	R/W	0h	Strobe event latch interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
8	PCM	R/W	0h	Position-compare match interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
7	PCR	R/W	0h	Position-compare ready interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
6	PCO	R/W	0h	Position counter overflow interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
5	PCU	R/W	0h	Position counter underflow interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
4	WTO	R/W	0h	Watchdog time out interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled

Table 31-22. QEINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	QDC	R/W	0h	Quadrature direction change interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
2	QPE	R/W	0h	Quadrature phase error interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
1	PCE	R/W	0h	Position counter error interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
0	RESERVED	R	0h	Reserved

31.12.2.17 QFLG Register (Offset = 32h) [Reset = 0000h]

QFLG is shown in Figure 31-42 and described in Table 31-23.

Return to the Summary Table.

QEP Interrupt Flag

Figure 31-42. QFLG Register

15	14	13	12	11	10	9	8
RESERVED			QMAE	UTO	IEL	SEL	PCM
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	INT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 31-23. QFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	QMAE	R	0h	QMA Error interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Interrupt was generated
11	UTO	R	0h	Unit time out interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Set by eQEP unit timer period match
10	IEL	R	0h	Index event latch interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set after latching the QPOSCNT to QPOSILAT
9	SEL	R	0h	Strobe event latch interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set after latching the QPOSCNT to QPOSSLAT
8	PCM	R	0h	eQEP compare match event interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set on position-compare match
7	PCR	R	0h	Position-compare ready interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set after transferring the shadow register value to the active position compare register
6	PCO	R	0h	Position counter overflow interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set on position counter overflow.
5	PCU	R	0h	Position counter underflow interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set on position counter underflow.
4	WTO	R	0h	Watchdog timeout interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Set by watchdog timeout

Table 31-23. QFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	QDC	R	0h	Quadrature direction change interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Interrupt was generated
2	PHE	R	0h	Quadrature phase error interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Set on simultaneous transition of QEPA and QEPB
1	PCE	R	0h	Position counter error interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Position counter error
0	INT	R	0h	Global interrupt status flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Interrupt was generated

31.12.2.18 QCLR Register (Offset = 34h) [Reset = 0000h]

QCLR is shown in Figure 31-43 and described in Table 31-24.

Return to the Summary Table.

QEP Interrupt Clear

Figure 31-43. QCLR Register

15	14	13	12	11	10	9	8
RESERVED			QMAE	UTO	IEL	SEL	PCM
R-0h			R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	INT
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 31-24. QCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	QMAE	R-0/W1S	0h	Clear QMA Error interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
11	UTO	R-0/W1S	0h	Clear unit time out interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
10	IEL	R-0/W1S	0h	Clear index event latch interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
9	SEL	R-0/W1S	0h	Clear strobe event latch interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
8	PCM	R-0/W1S	0h	Clear eQEP compare match event interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
7	PCR	R-0/W1S	0h	Clear position-compare ready interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
6	PCO	R-0/W1S	0h	Clear position counter overflow interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
5	PCU	R-0/W1S	0h	Clear position counter underflow interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
4	WTO	R-0/W1S	0h	Clear watchdog timeout interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag

Table 31-24. QCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	QDC	R-0/W1S	0h	Clear quadrature direction change interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
2	PHE	R-0/W1S	0h	Clear quadrature phase error interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
1	PCE	R-0/W1S	0h	Clear position counter error interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
0	INT	R-0/W1S	0h	Global interrupt clear flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag

31.12.2.19 QFRC Register (Offset = 36h) [Reset = 0000h]

QFRC is shown in Figure 31-44 and described in Table 31-25.

Return to the Summary Table.

QEP Interrupt Force

Figure 31-44. QFRC Register

15	14	13	12	11	10	9	8
RESERVED			QMAE	UTO	IEL	SEL	PCM
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 31-25. QFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	QMAE	R/W	0h	Force QMA error interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
11	UTO	R/W	0h	Force unit time out interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
10	IEL	R/W	0h	Force index event latch interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
9	SEL	R/W	0h	Force strobe event latch interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
8	PCM	R/W	0h	Force position-compare match interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
7	PCR	R/W	0h	Force position-compare ready interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
6	PCO	R/W	0h	Force position counter overflow interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
5	PCU	R/W	0h	Force position counter underflow interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
4	WTO	R/W	0h	Force watchdog time out interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt

Table 31-25. QFRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	QDC	R/W	0h	Force quadrature direction change interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
2	PHE	R/W	0h	Force quadrature phase error interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
1	PCE	R/W	0h	Force position counter error interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
0	RESERVED	R	0h	Reserved

31.12.2.20 QEPSTS Register (Offset = 38h) [Reset = 0000h]

QEPSTS is shown in Figure 31-45 and described in Table 31-26.

Return to the Summary Table.

QEP Status

Figure 31-45. QEPSTS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
UPEVNT	FIDF	QDF	QDLF	COEF	CDEF	FIMF	PCEF
R/W1C-0h	R-0h	R-0h	R-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R-0h

Table 31-26. QEPSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	UPEVNT	R/W1C	0h	Unit position event flag Reset type: SYSRSn 0h (R/W) = No unit position event detected 1h (R/W) = Unit position event detected. Write 1 to clear
6	FIDF	R	0h	Direction on the first index marker Status of the direction is latched on the first index event marker. Reset type: SYSRSn 0h (R/W) = Counter-clockwise rotation (or reverse movement) on the first index event 1h (R/W) = Clockwise rotation (or forward movement) on the first index event
5	QDF	R	0h	Quadrature direction flag Reset type: SYSRSn 0h (R/W) = Counter-clockwise rotation (or reverse movement) 1h (R/W) = Clockwise rotation (or forward movement)
4	QDLF	R	0h	eQEP direction latch flag Reset type: SYSRSn 0h (R/W) = Counter-clockwise rotation (or reverse movement) on index event marker 1h (R/W) = Clockwise rotation (or forward movement) on index event marker
3	COEF	R/W1C	0h	Capture overflow error flag Reset type: SYSRSn 0h (R/W) = Overflow has not occurred. 1h (R/W) = Overflow occurred in eQEP Capture timer (QEPCTMR). This bit is cleared by writing a '1'.
2	CDEF	R/W1C	0h	Capture direction error flag Reset type: SYSRSn 0h (R/W) = Capture direction error has not occurred. 1h (R/W) = Direction change occurred between the capture position event. This bit is cleared by writing a '1'.
1	FIMF	R/W1C	0h	First index marker flag Reset type: SYSRSn 0h (R/W) = First index pulse has not occurred. 1h (R/W) = Set by first occurrence of index pulse. This bit is cleared by writing a '1'.

Table 31-26. QEPSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	PCEF	R	0h	Position counter error flag. This bit is not sticky and it is updated for every index event. Reset type: SYSRSn 0h (R/W) = No error occurred during the last index transition 1h (R/W) = Position counter error

31.12.2.21 QCTMR Register (Offset = 3Ah) [Reset = 0000h]

QCTMR is shown in Figure 31-46 and described in Table 31-27.

Return to the Summary Table.

QEP Capture Timer

Figure 31-46. QCTMR Register

15	14	13	12	11	10	9	8
QCTMR							
R/W-0h							
7	6	5	4	3	2	1	0
QCTMR							
R/W-0h							

Table 31-27. QCTMR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QCTMR	R/W	0h	This register provides time base for edge capture unit. Reset type: SYSRSn

31.12.2.22 QCPRD Register (Offset = 3Ch) [Reset = 0000h]

QCPRD is shown in Figure 31-47 and described in Table 31-28.

Return to the Summary Table.

QEP Capture Period

Figure 31-47. QCPRD Register

15	14	13	12	11	10	9	8
QCPRD							
R/W-0h							
7	6	5	4	3	2	1	0
QCPRD							
R/W-0h							

Table 31-28. QCPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QCPRD	R/W	0h	This register holds the period count value between the last successive eQEP position events Reset type: SYSRSn

31.12.2.23 QCTMRLAT Register (Offset = 3Eh) [Reset = 0000h]

QCTMRLAT is shown in Figure 31-48 and described in Table 31-29.

Return to the Summary Table.

QEP Capture Latch

Figure 31-48. QCTMRLAT Register

15	14	13	12	11	10	9	8
QCTMRLAT							
R-0h							
7	6	5	4	3	2	1	0
QCTMRLAT							
R-0h							

Table 31-29. QCTMRLAT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QCTMRLAT	R	0h	The eQEP capture timer value can be latched into this register on two events viz., unit timeout event, reading the eQEP position counter. Reset type: SYSRSn

31.12.2.24 QCPRDLAT Register (Offset = 40h) [Reset = 0000h]

QCPRDLAT is shown in Figure 31-49 and described in Table 31-30.

Return to the Summary Table.

QEP Capture Period Latch

Figure 31-49. QCPRDLAT Register

15	14	13	12	11	10	9	8
QCPRDLAT							
R-0h							
7	6	5	4	3	2	1	0
QCPRDLAT							
R-0h							

Table 31-30. QCPRDLAT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QCPRDLAT	R	0h	eQEP capture period value can be latched into this register on two events viz., unit timeout event, reading the eQEP position counter. Reset type: SYSRSn

31.12.2.25 REV Register (Offset = 60h) [Reset = 00000009h]

REV is shown in Figure 31-50 and described in Table 31-31.

Return to the Summary Table.

QEP Revision Number

Figure 31-50. REV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										MINOR			MAJOR		
R-0-0h										R-1h			R-1h		

Table 31-31. REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5-3	MINOR	R	1h	This field specifies the Minor Revision number for the eQEP IP. Reset type: N/A
2-0	MAJOR	R	1h	This field specifies the Major Revision number for the eQEP IP. Reset type: N/A

31.12.2.26 QEPSTROBESEL Register (Offset = 64h) [Reset = 00000000h]

QEPSTROBESEL is shown in Figure 31-51 and described in Table 31-32.

Return to the Summary Table.

QEP Strobe select register

Figure 31-51. QEPSTROBESEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						STROBESEL	
R-0-0h						R/W-0h	

Table 31-32. QEPSTROBESEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1-0	STROBESEL	R/W	0h	Strobe source select: Reset type: SYSRSn 0h (R/W) = QEP Strobe after polarity mux 1h (R/W) = QEP Strobe after polarity mux 2h (R/W) = QEP Strobe after polarity mux ORed with ADCSOCA 3h (R/W) = QEP Strobe after polarity mux ORed with ADCSOCB

31.12.2.27 QMACTRL Register (Offset = 68h) [Reset = 00000000h]

QMACTRL is shown in Figure 31-52 and described in Table 31-33.

Return to the Summary Table.

QMA Control register

Figure 31-52. QMACTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													MODE		
R-0-0h													R/W-0h		

Table 31-33. QMACTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2-0	MODE	R/W	0h	Select Mode for QMA mode: 000 : QMA Module is bypassed. 001 : QMA Mode-1 operation selected 010 : QMA Mode-2 operation selected 011 : QMA Module is bypassed (reserved) 1xx : QMA Module is bypassed (reserved) Reset type: SYSRSn

31.12.2.28 QEPSRCSEL Register (Offset = 6Ch) [Reset = 00000000h]

QEPSRCSEL is shown in Figure 31-53 and described in Table 31-34.

Return to the Summary Table.

QEP Source Select Register

Figure 31-53. QEPSRCSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QEPSSEL				QEPISEL				QEPBSEL				QEPASEL			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 31-34. QEPSRCSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-12	QEPSSEL	R/W	0h	QEP Strobe source select: 0x0: From device Pins (Default). Others: Tied to zero. Reset type: SYSRSn
11-8	QEPISEL	R/W	0h	QEP Index source select: 0x0: Device Pin (Default) 0x1: CMPSS1.CTRIPH 0x2: CMPSS2.CTRIPH 0x3: CMPSS3.CTRIPH 0x4: CMPSS4.CTRIPH 0x5: CMPSS5.CTRIPH 0x6: CMPSS6.CTRIPH 0x7: CMPSS7.CTRIPH 0x8: CMPSS8.CTRIPH 0x9: PWMXBAR.1 0xA:PWMXBAR.2 0xB:PWMXBAR.3 0xC:PWMXBAR.4 0xD:PWMXBAR.5 0xE:PWMXBAR.6 0xF:PWMXBAR.7 Note: eQEP needs to be disabled before configuring these bits as it can lead to unexpected behavior if eQEP is running. Reset type: SYSRSn

Table 31-34. QEPSRCSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-4	QEPBSEL	R/W	0h	QEPB source select: 0x0: Device Pin (Default) 0x1: CMPSS1.CTRIPH 0x2: CMPSS2.CTRIPH 0x3: CMPSS3.CTRIPH 0x4: CMPSS4.CTRIPH 0x5: CMPSS5.CTRIPH 0x6: CMPSS6.CTRIPH 0x7: CMPSS7.CTRIPH 0x8: CMPSS8.CTRIPH 0x9: PWMXBAR.1 0xA:PWMXBAR.2 0xB:PWMXBAR.3 0xC:PWMXBAR.4 0xD:PWMXBAR.5 0xE:PWMXBAR.6 0xF:PWMXBAR.7 Note: eQEP needs to be disabled before configuring these bits as it can lead to unexpected behavior if eQEP is running. Reset type: SYSRSn
3-0	QEPASEL	R/W	0h	QEPA source select: 0x0: Device Pin (Default) 0x1: CMPSS1.CTRIPH 0x2: CMPSS2.CTRIPH 0x3: CMPSS3.CTRIPH 0x4: CMPSS4.CTRIPH 0x5: CMPSS5.CTRIPH 0x6: CMPSS6.CTRIPH 0x7: CMPSS7.CTRIPH 0x8: CMPSS8.CTRIPH 0x9: PWMXBAR.1 0xA:PWMXBAR.2 0xB:PWMXBAR.3 0xC:PWMXBAR.4 0xD:PWMXBAR.5 0xE:PWMXBAR.6 0xF:PWMXBAR.7 Note: eQEP needs to be disabled before configuring these bits as it can lead to unexpected behavior if eQEP is running. Reset type: SYSRSn

Chapter 32

Sigma Delta Filter Module (SDFM)

The sigma delta filter module (SDFM) is a four-channel digital filter designed specifically for current measurement and resolver position decoding in motor control applications. Each input channel can receive an independent delta-sigma (ΔΣ) modulator bit stream. The bit streams are processed by four individually-programmable digital decimation filters. The filter set includes a fast comparator (secondary filter) for immediate digital threshold comparisons for over-current and under-current monitoring, and zeros crossing detection.

32.1 Introduction

Figure 32-1 shows the SDFM CPU Interface.

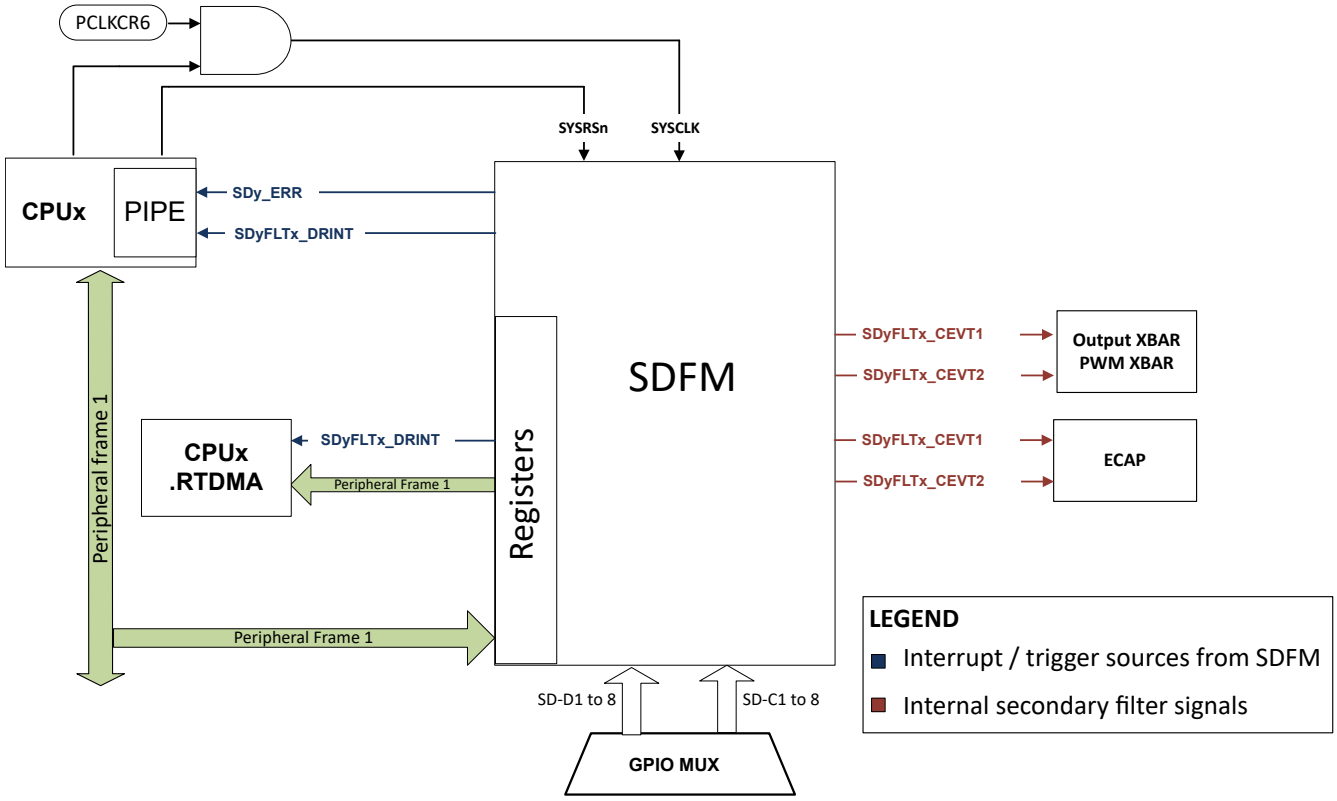


Figure 32-1. Sigma Delta Filter Module (SDFM) CPU Interface

32.1.1 SDFM Related Collateral

Foundational Materials

- C28x Academy - SDFM
- C29x Academy - Sigma-Delta Filter Module (SDFM)
- *How delta-sigma ADCs work, Part 1 Application Report*
- *How delta-sigma ADCs work, Part 2 Application Report*
- Nuts and Bolts of the Delta-Sigma Converter (Video)
- Sigma Delta Filter Module (SDFM) Training for C2000™ MCUs (Video)

Getting Started Materials

- *Achieving Better Signal Integrity With Isolated Delta-Sigma Modulators in Motor Drives Application Report*
  - Critical information for a hardware designer
- Using Sigma Delta Filter Module (SDFM) to Measure the Analog Input Signal
  - NOTE: This is a non-TI (third party) site.

Expert Materials

- C2000 DesignDRIVE Development Kit for Industrial Motor Control
- *Clock Edge Delay Compensation With Isolated Modulators Digital Interface to MCUs Application Report*
- *Diagnosing Delta-Sigma Modulator Bitstream Using C2000™ Configurable Logic Block Application Report*
- *Isolated Current Shunt and Voltage Measurement Kit Application Report*
- Isolated, Shunt-Based Current Sensing Reference Design

- The case for isolated delta-sigma modulators: Is my system fast enough for short-circuit detection?
- Vienna Rectifier-Based Three Phase Power Factor Correction Reference Design Using C2000 MCU

32.1.2 Features

SDFM features include:

- Eight external pins per SDFM module
  - Four sigma-delta data input pins per SDFM module (SD-Dx, where x = 1 to 4)
  - Four sigma-delta clock input pins per SDFM module (SD-Cx, where x = 1 to 4)
- Different configurable modulator clock modes supported:
  - Mode 0: Modulator clock rate equals the modulator data rate.
- Four independent, configurable secondary filter (comparator) units per SDFM module:
  - Four different filter type selection (Sinc1/Sinc2/SincFast/Sinc3) options available
  - Ability to detect over-value condition, under-value condition, and Threshold-crossing conditions
    1. Two independent Higher Threshold comparators (used to detect over-value condition)
    2. Two independent Lower Threshold comparators (used to detect under-value condition)
    3. One independent Threshold-Crossing comparator (used to measure duty cycle/frequency with eCAP)
  - OSR value for comparator filter unit (COSR) programmable from 1 to 32
- Four independent configurable primary filter (data filter) units per SDFM module:
  - Four different filter type selection (Sinc1/Sinc2/SincFast/Sinc3) options available
  - OSR value for data filter unit (DOSR) programmable from 1 to 256
  - Ability to enable or disable (or both) individual filter module
  - Ability to synchronize all four independent filters of an SDFM module by using the Main Filter Enable (MFE) bit or by using PWM signals
- Data filter output can be represented in either 16 bits or 32 bits.
- Data filter unit has a programmable mode FIFO to reduce interrupt overhead. The FIFO has the following features:
  - The primary filter (data filter) has a 16-deep x 32-bit FIFO.
  - The FIFO can interrupt the CPU after programmable number of data-ready events.
  - FIFO Wait-for-Sync feature: Ability to ignore data-ready events until the PWM synchronization signal (SDSYNC) is received. Once the SDSYNC event is received, the FIFO is populated on every data-ready event.
  - Data filter output can be represented in either 16 bits or 32 bits.
- PWMx.SOCA/SOCB can be configured to serve as SDSYNC source on a per-data-filter-channel basis.
- PWMs can be used to generate a modulator clock for sigma-delta modulators.
- Configurable Input Qualification available for both SD-Cx and SD-Dx
- Ability to use one filter channel clock (SD-C1) to provide clock to other filter clock channels.
- Configurable digital filter available on comparator filter events to blankout comparator events caused by spurious noise

32.1.3 Block Diagram

Each SDFM module has four independent filter modules. These filter modules are identical and can be configured independently. Each individual filter module has the following units:

- Input control unit
- Primary filter (data filter) unit
- Secondary filter (comparator filter) unit with 4 independent comparators

Figure 32-2 shows the SDFM module block diagram. The SDFM port operation is configured and controlled by the registers listed in Section 32.12.

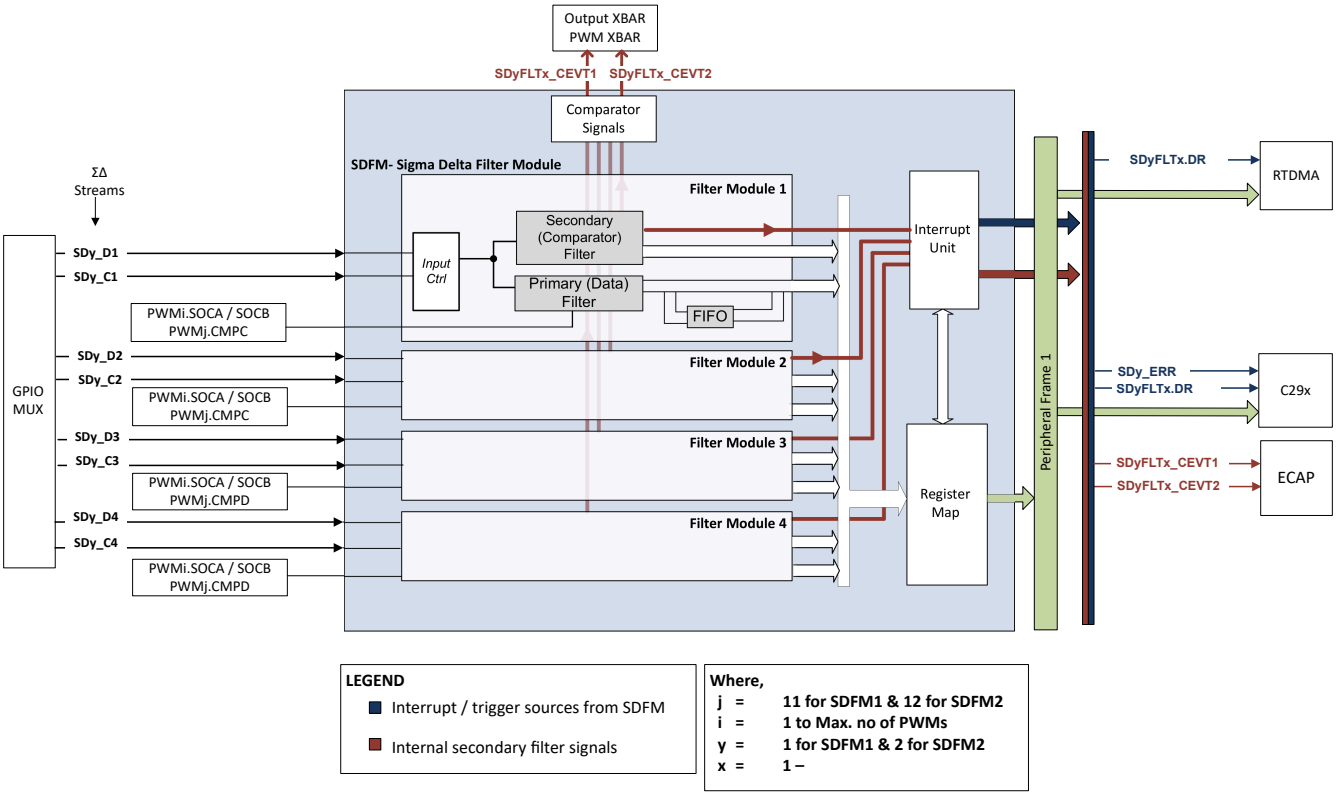


Figure 32-2. Sigma Delta Filter Module (SDFM) Block Diagram

Each filter module shown in Figure 32-3 has a primary (data) filter and a secondary (comparator) filter pair that receives the same bit stream. Except for the input bit stream, both the primary and secondary filter are completely independent of each other. Each of these filter modules can be independently configured. So, in a SDFM module, there is a total of four primary filters and four secondary filters.

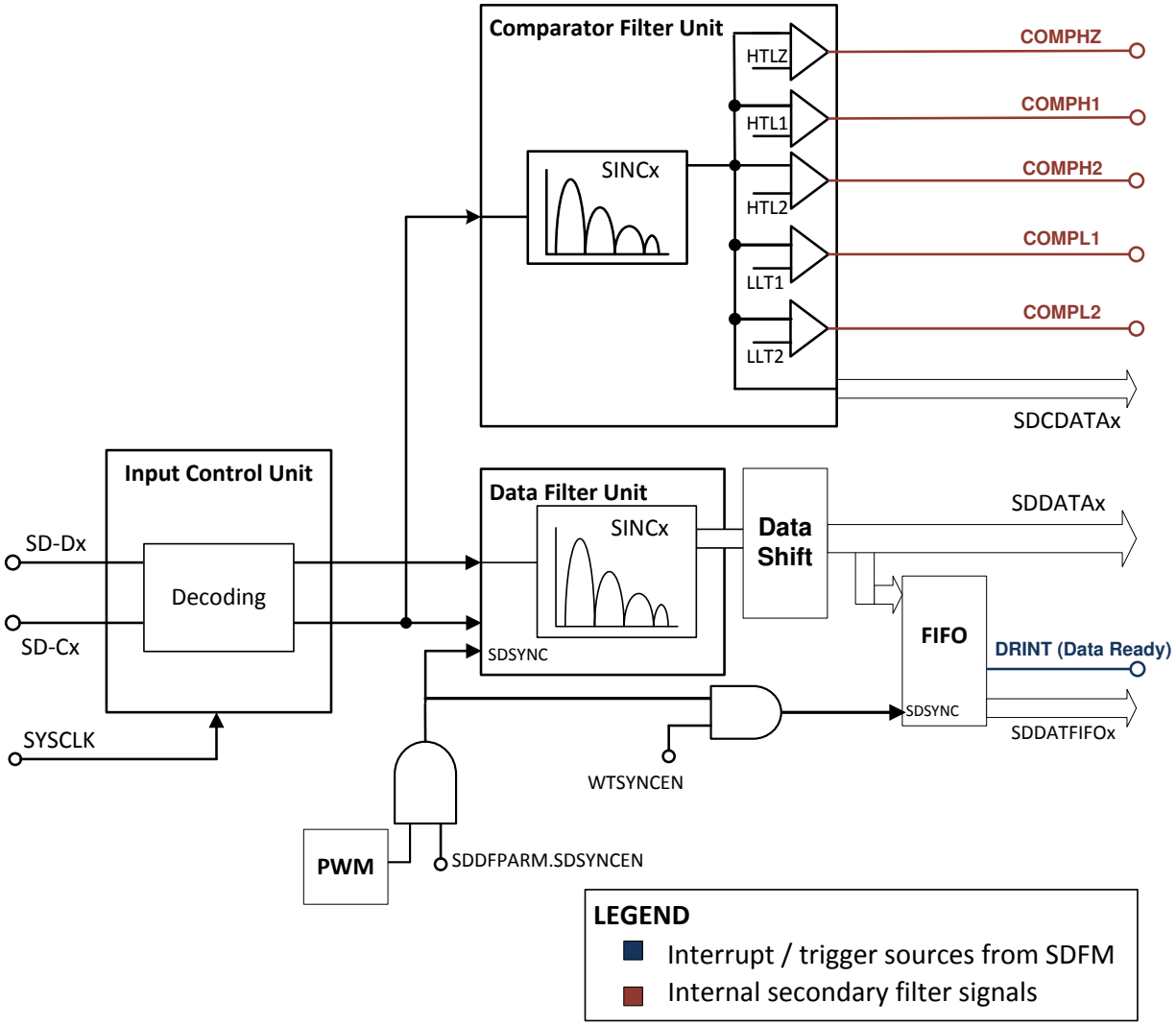


Figure 32-3. Block Diagram of One Filter Module

32.2 Configuring Device Pins

The GPIO mux registers must be configured to connect this peripheral to the device pins. To avoid glitches on the pins, the GPyGMUX bits must be configured first (while keeping the corresponding GPyMUX bits at the default of zero), followed by writing the GPyMUX register to the desired value.

For proper SDFM operation, use the following GPIO input qualification. Other GPIO qualifications are not supported.

- GPIO Input qualification is ASYNC, make sure to check the SDFM Electrical Data and Timing (Using ASYNC) requirement is met and be aware of the following caution message. SDFM Input Qualification feature is used to provide protection against random noise glitches.

CAUTION

The SDFM clock inputs (SDx_Cy pins) directly clock the SDFM module. Any glitches or ringing noise on these inputs can corrupt the SDFM module operation. Special precautions must be taken on these signals to make sure of a clean and noise-free signal that meets SDFM timing requirements. Precautions such as series termination for ringing due to any impedance mismatch of the clock driver and spacing of traces from other noisy signals are recommended.

Note

The SDFM module expects SD-Dx to change on the falling edge of SD-Cx and strobes for SD-Dx on the rising edge. But some SD-modulators in the market change SD-Dx on the rising edge and expect SDFM to strobe for data on the falling edge. In such cases, the GPIO inversion feature (GPxINV) is used on SD-Cx pin to change polarity and make it compatible with the SDFM.

See the *General-Purpose Input/Output (GPIO)* chapter for more details on GPIO mux and settings.

32.3 Input Qualification

Impulse noise sources such as EMI, crosstalk, and so on, has the possibility of corrupting SDCLK and SDDATA bit streams, resulting in corrupted SDFM filtered data. This impulse noise effects can be mitigated when using the input qualification feature that synchronizes SDCLK/SDDATA signals with PLLRAWCLK. By default, both SDCLK and SDDATA bit stream are not synchronized. SDCLK can be synchronized to PLLRAWCLK by setting SDCTLPARMx.SDCLKSYNC = 1 and SDDATA can be synchronized to PLLRAWCLK by setting SDCTLPARMx.SDDATASYNC = 1. Figure 32-4 shows optional Input Qualification option on SDCLK and SDDATA lines.

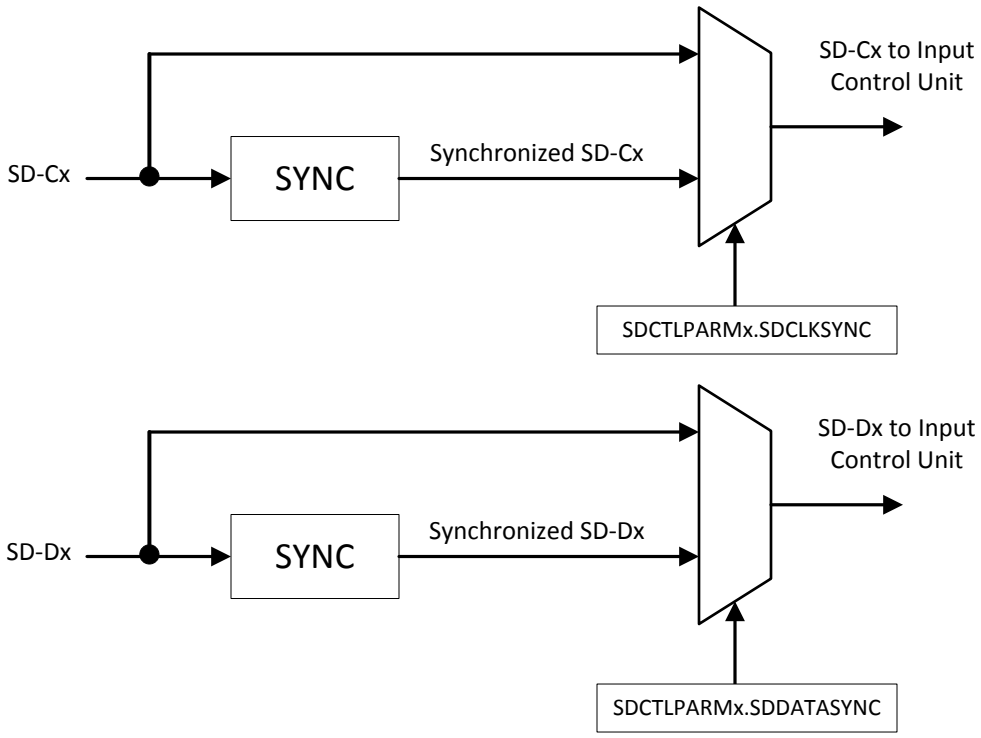


Figure 32-4. Input Qualification on SD-Cx and SD-Dx

Note

Using GPIO input synchronization when SDFM input qualification feature is enabled can cause unexpected results. The user must make sure that the GPIO pin is configured for asynchronous in this case.

32.4 Input Control Unit

The input control unit receives sigma delta modulated data and a sigma delta modulated clock. The modulated data received is captured and passed on to the data filter unit and comparator unit. This unit can be configured to receive the modulated data in Mode 0. Table 32-1 and Figure 32-5 show how SDCTLPARMx.MOD bits can be configured in Mode 0.

Table 32-1. Modulator Clock Modes

Modulator Mode [MOD]	Description
0	The modulator clock is running with the modulator data rate. The modulator data is strobed at every rising edge of the modulator clock.
1	Reserved
2	Reserved
3	Reserved

Note

To achieve the maximum value, the sigma-delta modulator has to be operated at absolute maximum positive or negative full scale, which is outside of the recommended full scale range of 80% of most sigma-delta modulators.

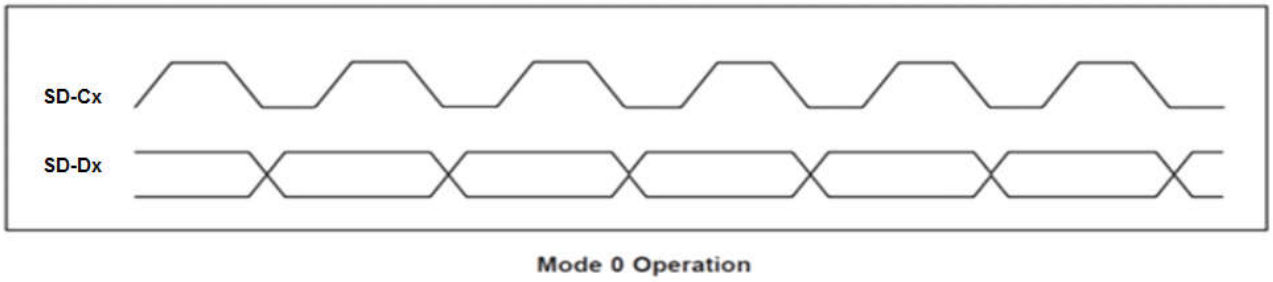


Figure 32-5. Different Modulator Modes Supported

32.5 SDFM Clock Control

In systems, the modulator clock can be generated using PWMs. Assuming all the SD-CLKs see the same delay on board traces, you can potentially use just one clock to clock multiple filters; thereby, saving on the number of pins used for SDFM. To enable this, Filter1 SDCLK (SD-C1) can possibly apply to other filter channels if required. The SDCTLPARAMx.SDCLKSEL register bit field can be configured to select filter channel SDCLK. See Figure 32-6 to view this feature.

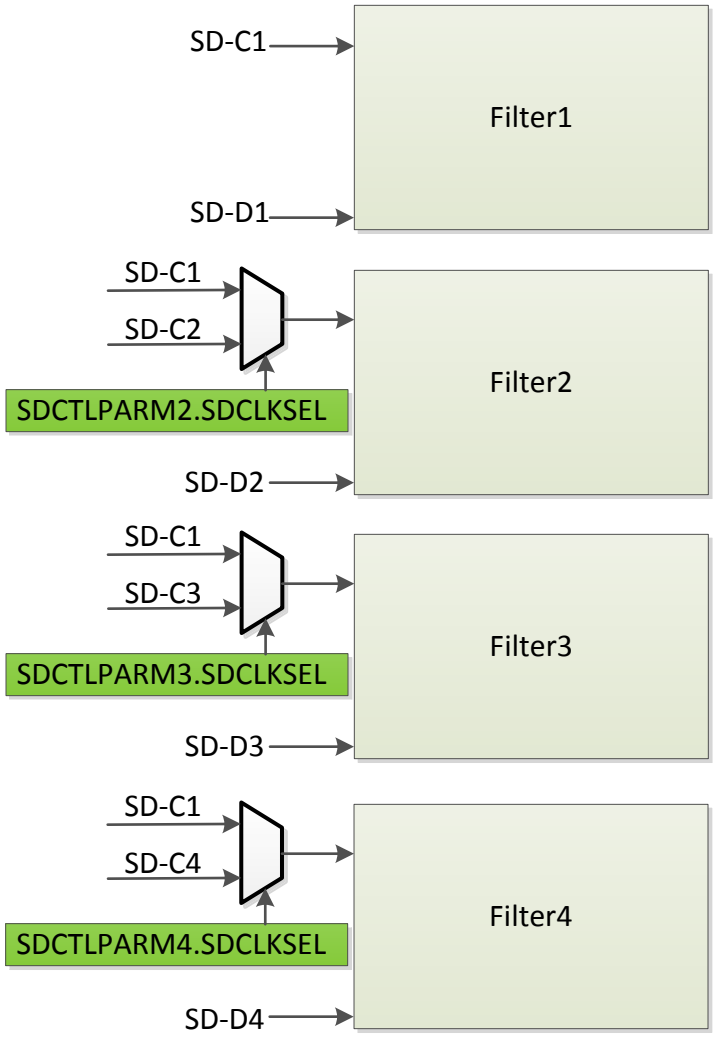


Figure 32-6. SDFM Clock Control

32.6 Sinc Filter

Both the comparator filter and data filter available in SDFM have the Sinc^N filter as the core. The Sinc^N filter is essentially a low-pass filter that converts the input bit stream into digital data by digital filtering and decimation. This filtered digital data represents analog input given to the sigma delta modulator. Simplified Sinc^N architecture consists of cascaded integrators and differentiators separated by a down-sampler as shown in Figure 32-7. The Z-transfer function of the Sinc filter of order N is shown in Figure 32-8.

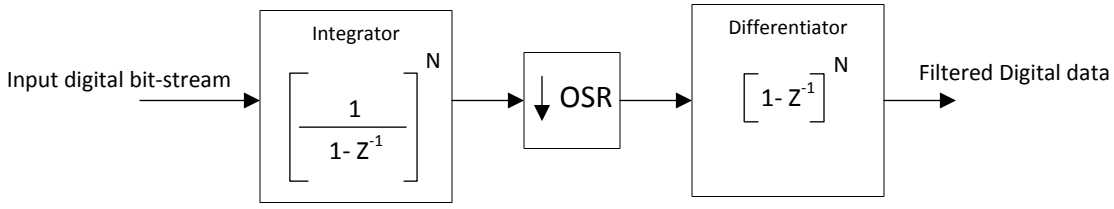


Figure 32-7. Simplified Sinc Filter Architecture

$$H(Z) = \left[\frac{1 - Z^{-OSR}}{1 - Z^{-1}} \right]^{N}$$

N = Order of Sinc filter
OSR = Over Sampling Ratio

Figure 32-8. Z-Transform of Sinc Filter of Order N

Effective resolution of the Sinc filter (ENOB) depends upon filter type, OSR and sigma-delta modulator frequency. Typically, higher resolution or ENOB can be achieved by higher OSR for a given filter type; however, the tradeoff is increased filter delay. It is important to choose the right sigma delta modulator by studying the optimal speed versus resolution tradeoff. Refer to the corresponding sigma delta modulator data sheet to determine the effective resolution for a given Sinc filter configuration. Figure 32-9 shows the frequency response of different filter structures when OSR = 32 and when the sigma delta modulator frequency is 10MHz.

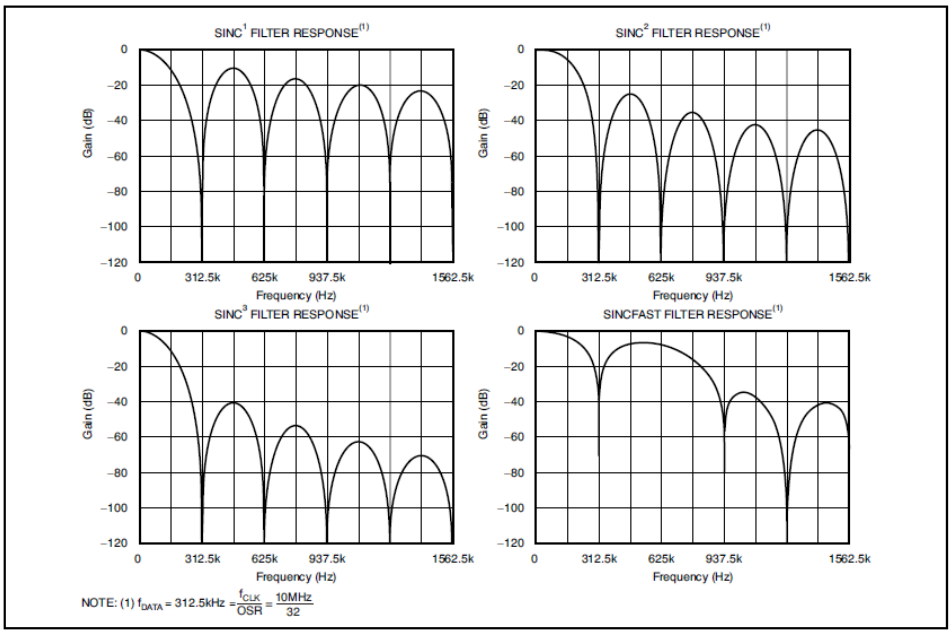


Figure 32-9. Frequency Response of Different Sinc Filters

The order of different sinc filter is shown in Table 32-2.

Table 32-2. Order of Sinc Filter

Filter Type	Order of Sinc Filter
Sinc1	1
Sinc2	2
Sinc3	3
SincFast	3

32.6.1 Data Rate and Latency of the Sinc Filter

The data rate of the sinc filter (filter throughput) represented in samples/sec is calculated by the following formula:

$$Data\ rate\ of\ Sinc\ filter = \frac{Modulator\ data\ rate}{OSR} \quad (28)$$

The latency of the sinc filter represented in secs is defined as the amount of time taken by a sinc filter type to deliver the correct filtered output upon initiation. For a given filter type, latency is calculated by the following formula:

$$Latency\ of\ Sinc\ filter = \frac{Order\ of\ Sinc\ filter}{Data\ rate\ of\ Sinc\ filter} \quad (29)$$

Example configuration:

Sinc filter type	= sinc3
Modulator data rate	= 10MHz
OSR	= 256
Data rate of Sinc Filter	= 10MHz/256 = 39.1K samples/second
Sinc filter latency	= 76.8µs

Sinc filter type	= sinc2
Modulator data rate	= 10MHz
OSR	= 256
Data rate of Sinc Filter	= 10MHz/256 = 39.1K samples/second
Sinc filter latency	=51.2µs

32.7 Data (Primary) Filter Unit

The data filter is a configurable Sinc filter which supports the following filter types: Sinc1, Sinc2, Sinc3, and SincFast. The data filter OSR (DOSR) settings can be configured from 1 to 256 and is independent of the comparator filter. Effective resolution of the data filter (ENOB) depends upon Data filter type, DOSR, and sigma-delta modulator frequency. By default, the data filter is disabled and setting of SDDFPARMx.FEN = 1 enables the data filter. The data filter output is represented in 26-bit signed integer in two's complement format. This filter unit translates a low input signal as '-1' and a high input signal as '1'. The resulting calculation gives both positive and negative values for the output of the data filter. Table 32-3 shows the different full scale values that the data filter can store using different OSRs.

See Section 32.6.1 to understand how to calculate data rate and latency of data filter.

Table 32-3. Peak Data Values for Different DOSR/Filter Combinations

DOSR	Sinc1	Sinc2	Sinc3	SincFast
x	x	x^2	x^3	$2x^2$
4	–4 to 4	–16 to 16	–64 to 64	–32 to 32
8	–8 to 8	–64 to 64	–512 to 512	–128 to 128
16	–16 to 16	–256 to 256	–4096 to 4096	–512 to 512
32	–32 to 32	–1024 to 1024	–32,768 to 32,768	–2048 to 2048
64	–64 to 64	–4096 to 4096	–262,144 to 262,144	–8192 to 8192
128	–128 to 128	–16,384 to 16,384	–2,097,152 to 2,097,152	–32,768 to 32,768
256	–256 to 256	–65,536 to 65,536	–16,777,216 to 16,777,216	–131,072 to 131,072

32.7.1 32-bit or 16-bit Data Filter Output Representation

The data filter output can be represented in either 32-bit or 16-bit format.

32-bit data filter representation:

- When SDDPARMx.DR = 1, data filter output is represented in 32-bit format. Writes to shift control bits do not have any bearing on the output of the data filter in this configuration.

16-bit data filter representation:

- By default, data filter output is represented in 16-bit format
- When SDDPARMx.DR = 0, data filter output is represented in 16-bit format. But it is the responsibility of the user to configure the corresponding shift control bits in the SDDPARMx register to control which 16-bit part of the 32-bit word is sent to the register map.

  For example, for the data filter configuration below:
  - Filter type = Sinc3
  - OSR = 128
  - SDDPARMx.DR = 0

  The data filter with a 26-bit signed output value can be in the range of –2,097,152 to 2,097,152. But, 16-bit signed output supports values only from –32,768 to 32,767. Therefore, it is required to configure shift control bits (SDDPARMx.SH) to 7 to represent the data filter output correctly in 16-bit format. Table 32-4 shows the configuration settings of shift control bits for different OSR and filter types.

Table 32-4. Shift Control Bit Configuration Settings

OSR	Sinc1	Sinc2	SincFast	Sinc3
1 to 31	0	0	0	0
32 to 40	0	0	0	1
41 to 50	0	0	0	2
51 to 63	0	0	0	3
64 to 80	0	0	0	4
81 to 101	0	0	0	5
102 to 127	0	0	0	6
128 to 161	0	0	1	7
162 to 181	0	0	1	8
182 to 203	0	1	2	8
204 to 255	0	1	2	9
256	0	2	3	10

CAUTION

Configuring shift control bits incorrectly results in getting an incorrect 16-bit data filter output.

32.7.2 Data FIFO

Each primary (data) filter channel has a 16-level deep, 32-bit FIFO.

FIFOs can be configured to collect a programmable number of data filter samples before issuing data-ready interrupt. This reduces the number of data-ready interrupts generated and resulting interrupt overhead for managed data flow.

By default, FIFO operation is disabled. FIFOs can be enabled by setting SDFIFOCTLx.FFEN = 1. When FIFO is enabled, each data-ready event from the data filter populates the FIFO, and the status of the FIFO at any given time is updated in the SDFIFOCTLx.SDFFST bit field.

Setting up FIFO to interrupt after receiving programmable number of data ready events:

- Enable SDFM FIFO (Set SDFIFOCTLx.FFEN = 1)
- Enable SDFM FIFO interrupt (Set SDFIFOCTLx.FFIEN = 1)
- Configure SDFIFOCTLx.SDFFIL bit field to any value between 0 to 16
- Configure SDFM data ready event to interrupt on FIFO interrupt (SDFFINT) (Set SDFFINTx = 1)
- Select data-ready interrupt source is SDFFINTx (DRINTx = SDFFINTx) (SDFIFOCTLx.DRINTSEL = 1)

When the SDFIFOCTLx.SDFFST >= SDFIFOCTLx.SDFFIL condition is met, the SDIFLG.SDFFINTx bit is set and an interrupt is generated on the DRINTx. SDIFLG.SDFFINTx flag can be cleared by setting the SDIFLGCLR. SDFFINTx bit field.

Wait for Sync feature:

The FIFO wait for sync feature can be used to ignore data-ready events from the data filter until the SDSYNC (from PWM) event is triggered.

By default, the Wait for Sync feature is disabled. This feature can be enabled by setting SDSYNCx.WTSYNCEN = 1

When the wait for sync feature is disabled:

FIFOs get populated on every data ready event until the FIFO gets full (or) when SDFIFOCTLx.SDFFST >= SDFIFOCTLx.SDFFIL.

When the wait for sync feature enabled:

FIFOs do not get populated on every data ready event until the FIFO receives a SDSYNC event. On a SYSYNC event, the FIFO sets SDSYNCx.WTSYNFLG = 1 and data ready events from the primary filter start populating the FIFO until either the FIFOs get full or when SDFIFOCTLx.SDFFST >= SDFIFOCTLx.SDFFIL. WTSYNFLG can be cleared either automatically or manually.

When WTSYNFLG = 0, FIFOs contents are frozen and subsequent data ready events do not populate FIFO until next SDSYNC event.

WTSYNFLG automatic clear mode:

By default, this mode is enabled. When SDSYNCx.WTSCLREN = 1, WTSYNFLG is automatically cleared on SDFFINT event.

WTSYNFLG manual clear mode:

Setting SDSYNCx.WTSYNCLR = 1 can be used to clear WTSYNFLG manually.

Clearing FIFO contents:

FIFO contents can cleared by any of the following methods:-

- Disabling FIFO clear FIFO contents. This can be done by clearing SDFIFOCTLx.FFEN = 0.
- Disabling Primary filter clear FIFO contents. This can done by either clearing SDDFPARMx.FEN = 0 (or) by clearing SDMFILEN.MFE = 0.
- FIFO contents can also be automatically cleared upon receiving the SDSYNC event. By default, this feature is disabled and this feature can be enabled by setting FIFO Clear-on-SDSYNC enable (SDSYNCx.FFSYNCCLREN = 1).

  Note: The above feature is only enabled when wait for sync feature is enabled (SDSYNCx. WTSYNCEN = 1).

FIFO debug access behavior:

Debug access of the SDDATFIFOx registers does not affect the FIFO pointers. On a CPU/RTDMA access to the SDDATFIFOx register, the FIFO read pointers advance to the next available entry in the FIFO.

32.7.3 SDSYNC Event

Primary (data) filters can be synchronized with respect to the PWM event (called SDSYNC event). The SDSYNC signal from the PWM module is used to reset the DOSR counter. This feature is by default disabled and can be enabled by setting SDDFPARMx.SDSYNCEN = 1. Each primary filter can be synchronized from any of the available PWMx SOCA/SOCB signals (see Table 32-5). Additionally, PWM11.CMPC/CMPD can be used to reset SDFM1 filter modules and PWM12.CMPC/CMPD can be used to reset SDFM2 filter modules. The SDSYNCx.SDSYNCSEL bits allow the user to configure which PWM signal provides the SDSYNC pulse to the primary filter. Figure 32-10 shows how device PWM signals are connected to the SDFM modules.

Table 32-5. SDSYNCx.SYNCSEL

SDSYNCxSYNCSEL	Input Signal
0	EPWM1_SOCA
1	EPWM1_SOCB
2	EPWM17_SOCA
3	EPWM17_SOCB
4	EPWM2_SOCA
5	EPWM2_SOCB
6	EPWM18_SOCA
7	EPWM18_SOCB
8	EPWM3_SOCA
9	EPWM3_SOCB
10-11	Reserved
12	EPWM4_SOCA
13	EPWM4_SOCB
14-15	Reserved
16	EPWM5_SOCA
17	EPWM5_SOCB
18-19	Reserved
20	EPWM6_SOCA
21	EPWM6_SOCB
22-23	Reserved
24	EPWM7_SOCA
25	EPWM7_SOCB
26-27	Reserved
28	EPWM8_SOCA
29	EPWM8_SOCB
30-31	Reserved
32	EPWM9_SOCA
33	EPWM9_SOCB
34-35	Reserved
36	EPWM10_SOCA
37	EPWM10_SOCB

Table 32-5. SDSYNCx.SYNCSEL (continued)

SDSYNCxSYNCSEL	Input Signal
38-39	Reserved
40	EPWM11_SOCA
41	EPWM11_SOCB
42-43	Reserved
44	EPWM12_SOCA
45	EPWM12_SOCB
46-47	Reserved
48	EPWM13_SOCA
49	EPWM13_SOCB
50-51	Reserved
52	EPWM14_SOCA
53	EPWM14_SOCB
54-55	Reserved
56	EPWM15_SOCA
57	EPWM15_SOCB
58-59	Reserved
60	EPWM16_SOCA
61	EPWM16_SOCB
62	Reserved
63	SDFM1.SDSYNC1 - EPWM11_CTR_CMPC SDFM1.SDSYNC2 - EPWM11_CTR_CMPC SDFM1.SDSYNC3 - EPWM11_CTR_CMPD SDFM1.SDSYNC4 - EPWM11_CTR_CMPD SDFM2.SDSYNC1 - EPWM12_CTR_CMPC SDFM2.SDSYNC2 - EPWM12_CTR_CMPC SDFM2.SDSYNC3 - EPWM12_CTR_CMPD SDFM2.SDSYNC4 - EPWM12_CTR_CMPD SDFM3.SDSYNC1 - EPWM13_CTR_CMPC SDFM3.SDSYNC2 - EPWM13_CTR_CMPC SDFM3.SDSYNC3 - EPWM13_CTR_CMPD SDFM3.SDSYNC4 - EPWM13_CTR_CMPD SDFM4.SDSYNC1 - EPWM14_CTR_CMPC SDFM4.SDSYNC2 - EPWM14_CTR_CMPC SDFM4.SDSYNC3 - EPWM14_CTR_CMPD SDFM4.SDSYNC4 - EPWM14_CTR_CMPD

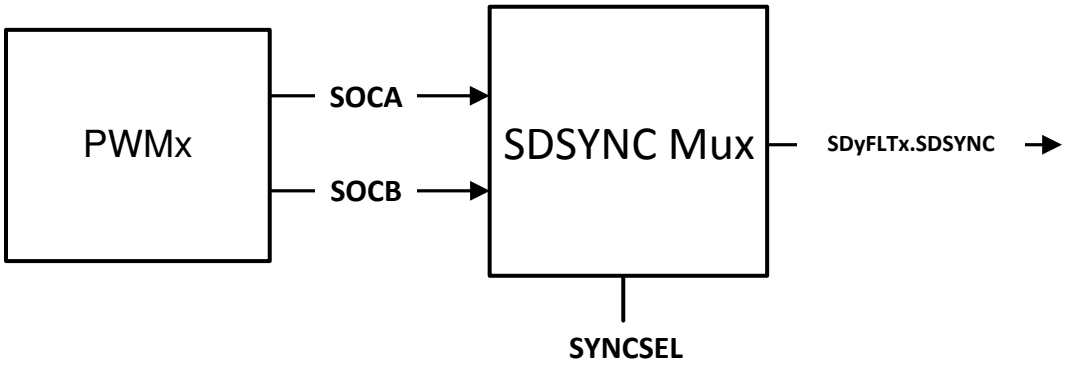


Figure 32-10. SDSYNC Event

Because of the inherent architecture of the Sinc filter (Sinc1, Sinc2, Sinc3, SincFast), the first few samples, depending upon filter type, are incorrect. Table 32-6 shows the number of incorrect samples on the following conditions:

- When Sinc filter is enabled and configured for first time.
- When Sinc filter is disabled and re-enabled or reconfigured in the middle of operation.
- When data filter receives SDSYNC event from PWM.

Table 32-6. Number of Incorrect Samples Tabulated

Filter Type	Number of Incorrect Samples After the Filter is Enabled and Configured
Sinc1	No incorrect sample.
Sinc2	The first sample of the Sinc2 filter is incorrect.
SincFast	The first two samples of the SincFast filter are incorrect.
Sinc3	The first two samples of the Sinc3 filter are incorrect.

CAUTION

SDFM comparator interrupts can be enabled only after providing sufficient settling time to make sure the comparator filter does not trip on these incorrect samples. Therefore, SDFM comparator interrupts (CEVT1 and CEVT2) can be enabled only after a sufficient delay is provided after the comparator filter is configured. This sufficient delay is calculated by adding the latency of the comparator filter and 5 SD-Cx clock cycles.

32.8 Comparator (Secondary) Filter Unit

Most control systems require protection of the system by tripping the PWM in case the current or voltage goes out of bounds. The primary purpose of the secondary (comparator) filter is to allow the user to monitor input conditions with a fast settling time. This allows the user to trip PWMs to protect the system from potential damage.

Note

The secondary (comparator) filter cannot be synchronized with respect to the PWM event (SDSYNC event).

The comparator filter is a configurable Sinc filter that supports the following filter types: Sinc1, Sinc2, Sinc3, and SincFast. The comparator OSR (COSR) settings can be configured from 1 to 32 and is independent of the data filter. Effective resolution of the comparator filter (ENOB) depends upon the comparator filter type, COSR, and sigma-delta modulator frequency. By default, the comparator filter is disabled and setting SDCPARMx.CEN = 1 enables the comparator filter. The comparator filter output is represented in 16-bit unsigned format. This filter unit translates a low input signal as 0 and a high input signal as 1. The resulting calculations give only positive values for the output of the comparator filter. Table 32-7 shows the different full-scale values that the comparator filter can store using different OSRs.

Table 32-7. Peak Data Values for Different OSR/Filter Combinations

OSR	Sinc1	Sinc2	Sinc3	SincFast
x	0 to x	0 to x2	0 to x3	0 to 2x2
4	0 to 4	0 to 16	0 to 64	0 to 32
8	0 to 8	0 to 64	0 to 512	0 to 128
16	0 to 16	0 to 256	0 to 4096	0 to 512
32	0 to 32	0 to 1024	0 to 32,768	0 to 2048

See Section 32.6.1 to understand how to calculate data rate and latency of comparator filter.

The output of the comparator filter is memory-mapped and can be read in the SDCDATAx register. This register, SDCDATAx, is updated every COSR number of SD-Cx cycles. The comparator filter digital output is connected to digital comparators explained below.

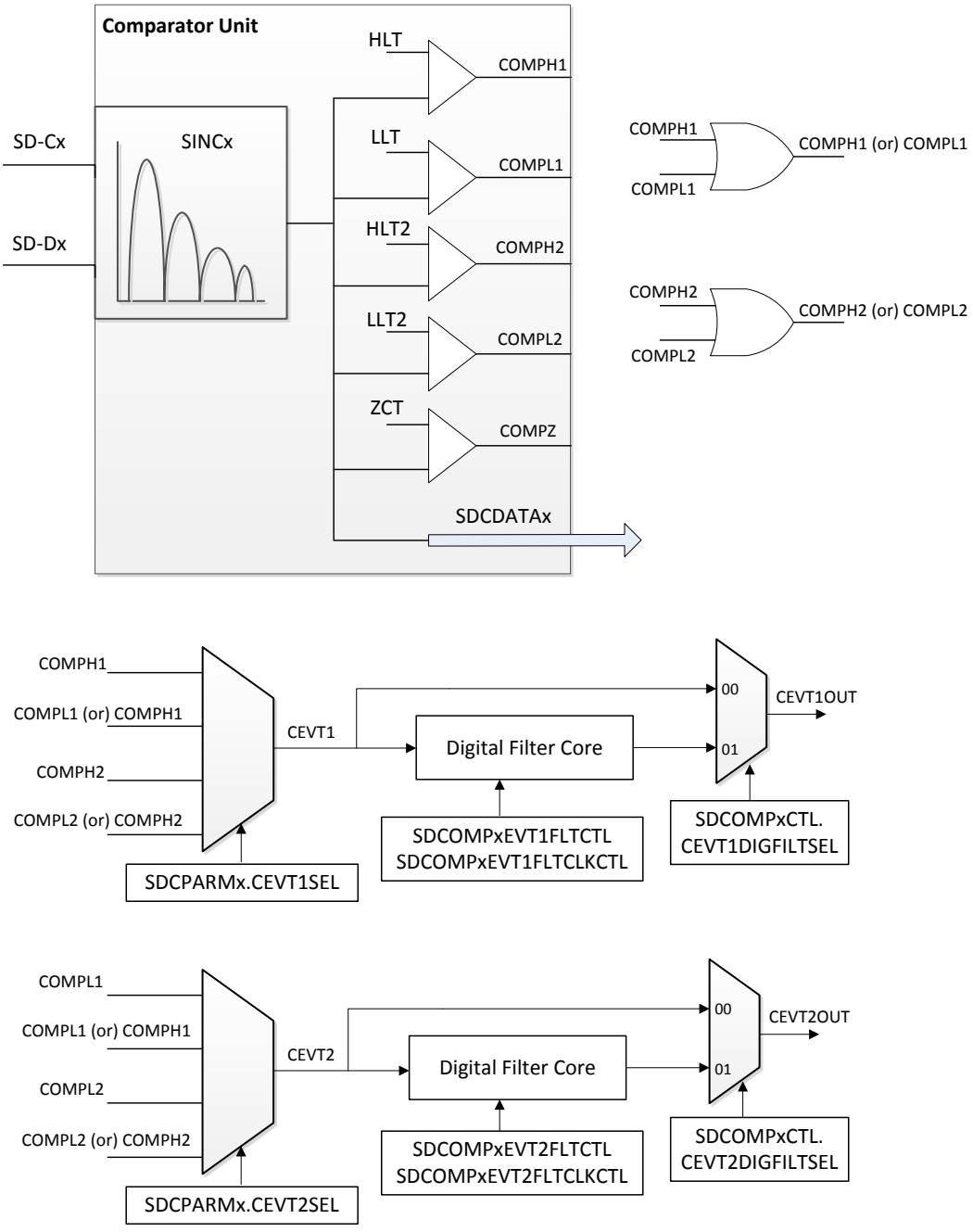


Figure 32-11. Comparator Unit Structure

32.8.1 Higher Threshold (HLT) Comparators

- High threshold comparator can be used to detect over-value condition.
- When comparator data > = higher threshold register, a high threshold event is generated.
- Higher threshold comparator events except for COMPHZx can be configured to trigger following events: CPU interrupt, PWM trip.
- This device has three high threshold comparators:
  - **Higher Threshold 1 (HLT1) Comparator**:
    - When comparator data > = (SDFLTxCMPH1.HLT), HLT1 comparator generates COMPH1 event.
    - The COMPH1 event is connected to both CEVT1 and CEVT2.
  - **Higher Threshold 2 (HLT2) Comparator**:
    - When comparator data > = (SDFLTxCMPH2.HLT), HLT2 comparator generates COMPH2 event.
    - The COMPH2 event is connected to both CEVT1 and CEVT2.
  - **Higher Threshold (HTLZ) Comparator**:
    - When comparator data > = (SDFLT1CMPHZ.CMPHZ), it can generate a Higher Threshold (B) event (COMPHZx) and sets the corresponding SDSTATUS.HZx flag. But, this event cannot be configured to generate SDFM interrupt (SDx_ERR). The COMPHZ signals from HTLZ comparator are connected to CLB XBAR.

32.8.2 Lower Threshold (LLT) Comparators

- The low threshold comparator can be used to detect under-value condition.
- When comparator data < = Lower Threshold register, a low threshold event is generated.
- Lower threshold comparator events can be configured to trigger following events: CPU interrupt, PWM trip.
- Lower threshold comparator events can be used in conjunction with ECAP to measure the frequency / duty cycle of Threshold crossing
- This device has two low threshold comparators. .
  - **Lower Threshold 1 (LLT1) Comparator**
    - When comparator data < = (SDFLTxCMPL1.LLT), the LLT1 comparator generates COMPL1 event.
    - The COMPL1 event is connected to both CEVT1 and CEVT2.
  - **Lower Threshold 2 (LLT2) Comparator**
    - When comparator data < = (SDFLTxCMPL2.LLT), LLT2 comparator generates COMPL2 event.
    - The COMPL2 event is connected to both CEVT1 and CEVT2.

32.8.3 Digital Filter

The digital filter works on a window of FIFO samples (SAMPWIN + 1) taken from the input. The filter output resolves to the majority value of the sample window, where majority is defined by the threshold (THRESH) value. If the majority threshold is not satisfied, the filter output remains unchanged.

For proper operation, the value of THRESH must be greater than SAMPWIN / 2.

A prescale function (CLKPRESCALE) determines the filter sampling rate, where the filter FIFO captures one sample every CLKPRESCALE system clocks. Old data from the FIFO is discarded.

A conceptual model of the digital filter is shown in Figure 32-12.

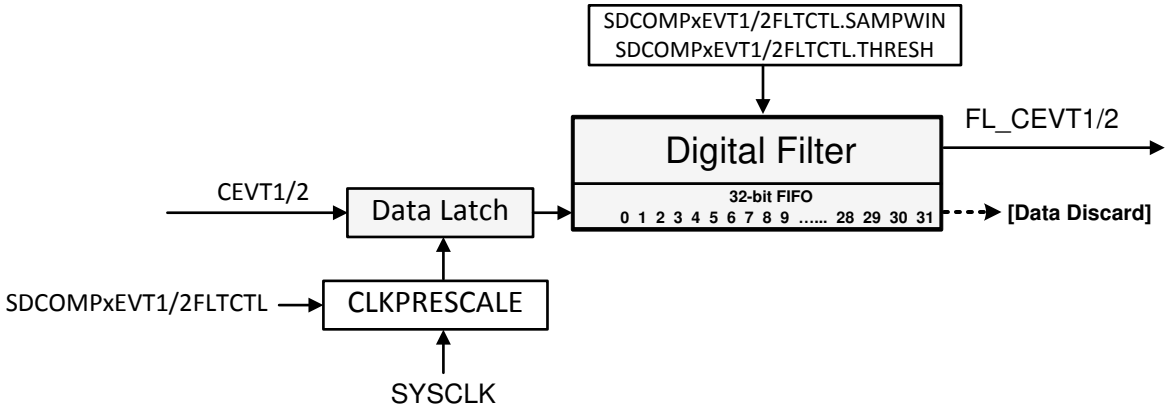


Figure 32-12. Digital Filter

Equivalent C code of the filter implementation is:

```
if (FILTER_OUTPUT == 0) {
    if (Num_1s_in_SAMPWIN >= THRESH) {
        FILTER_OUTPUT = 1;
    }
}
else {
    if (Num_0s_in_SAMPWIN >= THRESH) {
        FILTER_OUTPUT = 0;
    }
}
```

Filter Initialization Sequence

To make sure of proper operation of the digital filter, the following initialization sequence is recommended:

1. Configure the digital filter parameters for operation:
   - Set SAMPWIN for the number of samples to monitor in the FIFO window.
   - Set THRESH for the threshold required for majority qualification.
   - Set CLKPRESCALE for the digital filter clock prescale value.
2. Initialize the sample values in the digital FIFO window by setting FILINIT = 1.

32.9 Theoretical SDFM Filter Output

The following equations can be used to derive a theoretical filter output of an SDFM filter output for both a comparator filter and a data filter.

$$\text{Density of ones in bitstream} = \frac{\text{Input Voltage} + \text{Vclipping}}{2 \times \text{Vclipping}} \tag{30}$$

Where:

- Vclipping = maximum differential voltage input range of modulator
- Input voltage = Differential input voltage applied to the modulator

$$\text{Comparator Filter Output (Theoretical)} = \text{Density of ones in bitstream} \times \text{Maximum Filter Output (FilterType, COSR)} \tag{31}$$

$$\text{FilterOutput} = \left\{\frac{\text{absolute(Input voltage)}}{\text{Vclipping}}\right\} \times \text{Maximum Filter Output (FilterType, DOSR)} \tag{32}$$

$$\text{Data Filter Output\_32bit(Theoretical)} = \begin{cases} \text{FilterOutput} & \text{if Input Voltage is +ve voltage} \\ \text{2's complement of FilterOutput} & \text{if input voltage is -ve voltage} \end{cases} \tag{33}$$

$$\text{Data Filter Output\_16bit(Theoretical)} = \text{Data Filter Output\_32bit(Theoretical)} >> \text{Shift value(FilterType, OSR)} \tag{34}$$

For example, when using the AMC1306x25 modulator:

AMC1306x25	Vclipping = Input voltage (AINP - AINN) =	320mV 100mV
SDFM filter settings	Filter type = Comparator OSR (COSR) = Data filter OSR (DOSR) =	3 32 100

Density of 1s in bit stream	Using Equation 30	0.65625
Comparator filter output Filter type = Sinc3 COSR = 32	Using Equation 31	21504
Data filter output (32-bit) Filter type = Sinc3 DOSR = 100	Using Equation 32 and Equation 33	312500
Data filter output (32-bit) Filter type = Sinc3 DOSR = 100 (Right shift by 5)	Using Equation 34	9765

32.10 Interrupt Unit

Each SDFM can generate five CPU interrupts such as SDFM Error (SDy_ERR) and SDFM data ready (SDy_DRINT1 / SDy_DRINT2, SDy_DRINT3, SDy_DRINT4) interrupts for each filter module.

32.10.1 SDFM (SDyERR) Interrupt Sources

Figure 32-13 shows the structure of SDy_ERR interrupt. SDy_ERR interrupt can be triggered by any of these 16 events.

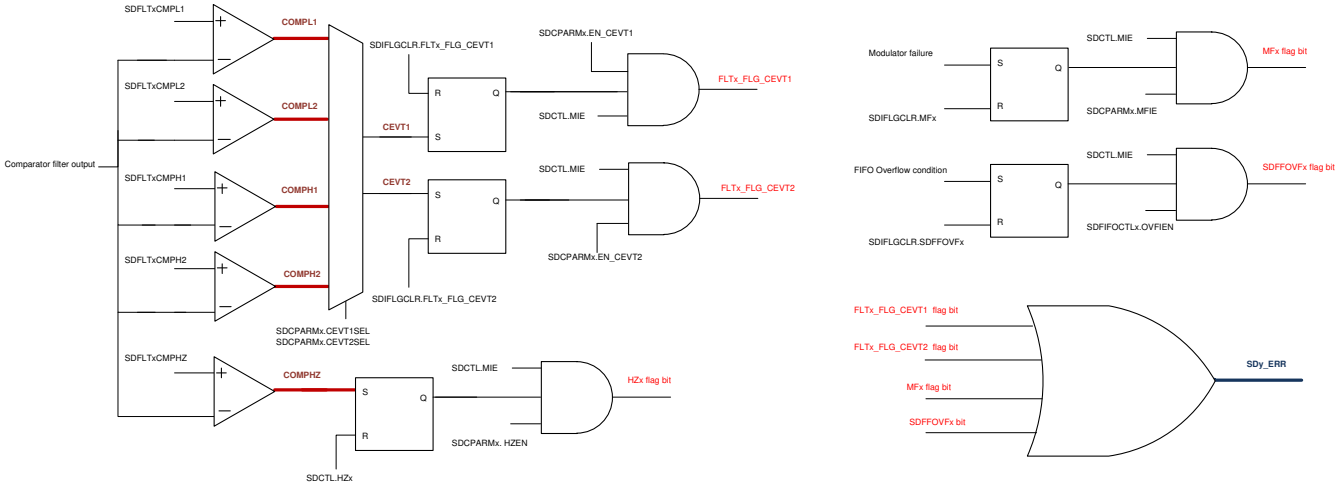


Figure 32-13. SDFM Error (SD_ERR) Interrupt Sources

1. Comparator Event1 (CEVT1)

CEVT1 events from any of the four comparator filter module can trigger CPU interrupt. This event can be configured to trigger SDy_ERR interrupt only if below configurations are made:

- Enable Main interrupt enable (SDCTL.MIE = 1)
- Enable comparator Event1 interrupt (SDCPARMx.EN_CEVT1 = 1)

On a CEVT1 event, SDIFLG.FLTx_FLG_CEVT1 flag bit is set. This flag bit can only be reset if the corresponding bit in SDIFLGCLR register is set and if the interrupt source is no longer active.

2. Comparator Event2 (CEVT2)

CEVT2 events from any of the four comparator filter module can trigger CPU interrupt. This event can be configured to trigger SDy_ERR interrupt only if below configurations are made:

- Enable Main interrupt enable (SDCTL.MIE = 1)
- Enable comparator event1 interrupt (SDCPARMx.EN_CEVT2 = 1)

On a CEVT2 event, SDIFLG.FLTx_FLG_CEVT2 flag bit is set. This flag bit can only be reset if the corresponding bit in SDIFLGCLR register is set and if the interrupt source is no longer active.

3. Modulator Failure (MFx) event

Modulator failures (MFx) are generated when SD-Cx goes missing. The modulator clock is considered missing if SD-Cx does not toggle for 64-SYSCLKs. MFx events from any of the four filter modules can trigger CPU interrupt. This event can be configured to trigger SDy_ERR interrupt only if below configurations are made:

- Enable Main Interrupt Enable (SDCTL.MIE = 1)
- Enable modulator clock failure interrupt source (SDCPARMx.MFIE = 1)

On a MFx event, SDIFLG.MFx flag bit is set. This flag bit can only be reset if the corresponding bit in SDIFLGCLR register is set and if the interrupt source is no longer active.

4. FIFO overflow (SDFFOVFx) event

The number of filter data available in FIFO at any given point can be tracked in SDFIFOCTLx.SDFFST. If the number of words received in FIFO is greater than Max FIFO depth (16), SDFFOVFx event is generated. SDFFOVFx events from any of the four filter modules can trigger CPU interrupt. This event can be configured to trigger SDy_ERR interrupt, only if below configurations are made:

- Enable SDFM FIFO (Set SDFIFOCTLx.FFEN = 1)
- Enable SDFM FIFO overflow interrupt (Set SDFIFOCTLx.OVFIEN = 1) and
- Enable Main interrupt enable (Set SDCTL.MIE = 1)

On a SDFFOVFx event, all subsequent data (primary) filter data is lost and is not stored in FIFO. SDIFLG.SDFFOVFx flag bit is set on a FIFO overflow event and this bit can be cleared if the corresponding bit in SDIFLGCLR register is set and if the interrupt source is no longer active.

32.10.2 Data Ready (DRINT) Interrupt Sources

Figure 32-14 shows the structure of interrupt SDy_DRINTx interrupt. Each SDy_DRINTx interrupt is triggered by corresponding Data Filter channel.

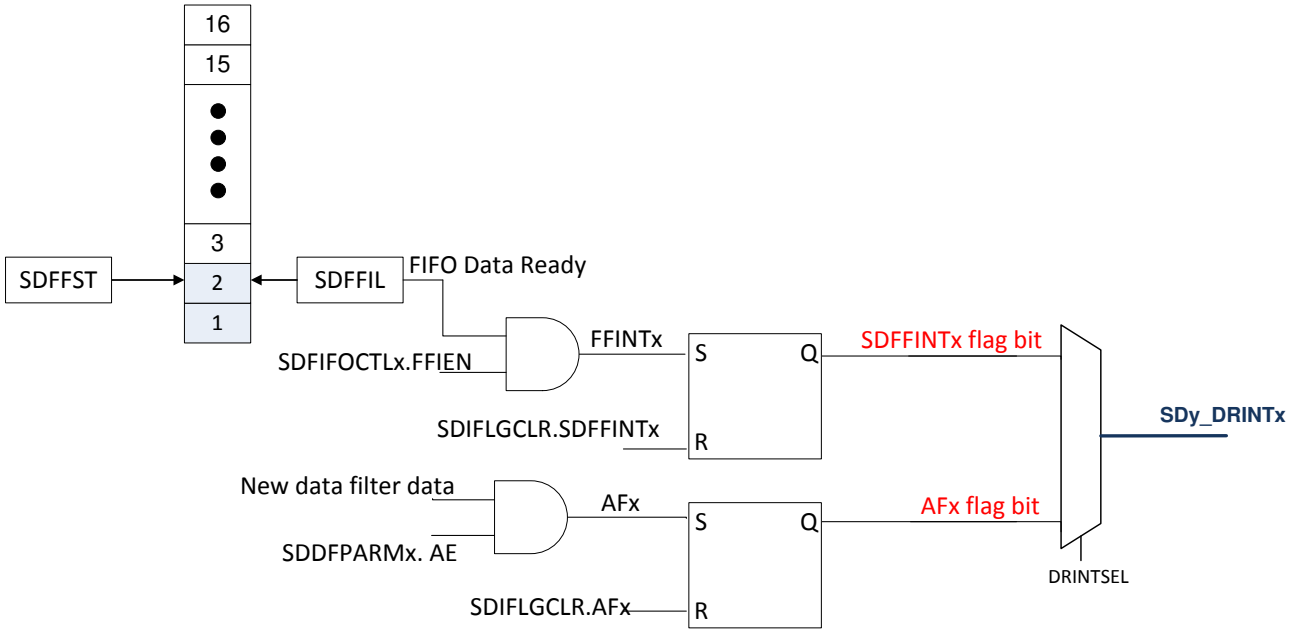


Figure 32-14. SDFM Data Ready (SDy_DRINTx) Interrupt

1. Data Acknowledge (AFx)

When the primary filter is ready with a new filter data, AFx event is generated. AFx events from each filter can generate an SDy_DRINTx interrupt. This event can be configured to trigger SDy_DRINTx interrupt only if below configurations are made:

- Enable individual filter interrupts (SDDFPARMx. AE = 1)
- Select data-ready interrupt source AFx (DRINTx = AFx) (SDFIFOCTLx.DRINTSEL = 0)

On an AFx event, the SDIFLG.AFx flag bit is set. This flag bit can only be reset, if the corresponding bit in SDIFLGCLR register is set and if the interrupt source is no longer active.

2. Four FIFO Data ready interrupt (SDFFINTx)

FIFO Data Ready event is generated whenever SDFIFOCTLx.SDFFST >= SDFIFOCTLx.SDFFIL condition is met. FIFO data ready events from each filter can generate an SDy_DRINTx interrupt. This event can be configured to trigger SDy_DRINTx interrupt only if below configurations are made:

Table 32-8 shows how the DRINTx output is selected.

- Enable SDFM FIFO (Set SDFIFOCTLx.FFEN = 1) and
- Enable SDFM FIFO interrupt (Set SDFIFOCTLx.FFIEN = 1)
- Select data-Ready interrupt source is SDFFINTx (DRINTx = SDFFINTx) (SDFIFOCTLx.DRINTSEL = 1)

Table 32-8. SDFM Data-Ready Interrupt (SDy_DRINTx) Output Selection

DRINTSEL	AE	FFIEN	FFEN	DRINTx
0	0	x	X	0
0	1	x	X	AFx
1	x	0	X	0
1	x	x	0	0
1	x	1	1	SDFFINTx

32.11 Software

32.11.1 SDFM Registers to Driverlib Functions

Table 32-9. SDFM Registers to Driverlib Functions

File	Driverlib Function
SDIFLG	
-	
SDIFLGCLR	
-	
SDCTL	
-	
SDMFILEN	
-	
SDSTATUS	
-	
SDINTMODE	
-	
SDCTLPARM1	
-	
SDDFPARM1	
-	
SDDPARM1	
-	
SDFLT1CMPH1	
-	
SDFLT1CMPL1	
-	
SDCPARM1	
-	
SDDATA1	
-	
SDDATFIFO1	
-	
SDCDATA1	
-	
SDFLT1CMPH2	
-	
SDFLT1CMPHZ	
-	
SDFIFOCTL1	
-	
SDSYNC1	
-	
SDFLT1CMPL2	
-	
SDCTLPARM2	
-	See SDCTLPARM1

Table 32-9. SDFM Registers to Driverlib Functions (continued)

File	Driverlib Function
SDDFPARM2	
-	See SDDFPARM1
SDDPARM2	
-	See SDDPARM1
SDFLT2CMPH1	
-	
SDFLT2CMPL1	
-	
SDCPARM2	
-	See SDCPARM1
SDDATA2	
-	See SDDATA1
SDDATFIFO2	
-	
SDCDATA2	
-	
SDFLT2CMPH2	
-	
SDFLT2CMPHZ	
-	
SDFIFOCTL2	
-	
SDSYNC2	
-	
SDFLT2CMPL2	
-	
SDCTLPARM3	
-	See SDCTLPARM1
SDDFPARM3	
-	See SDDFPARM1
SDDPARM3	
-	See SDDPARM1
SDFLT3CMPH1	
-	
SDFLT3CMPL1	
-	
SDCPARM3	
-	See SDCPARM1
SDDATA3	
-	See SDDATA1
SDDATFIFO3	
-	
SDCDATA3	
-	
SDFLT3CMPH2	

Table 32-9. SDFM Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
SDFLT3CMPHZ	
-	
SDFIFOCTL3	
-	
SDSYNC3	
-	
SDFLT3CMPL2	
-	
SDCTLPARM4	
-	See SDCTLPARM1
SDDFPARM4	
-	See SDDFPARM1
SDDPARM4	
-	See SDDPARM1
SDFLT4CMPH1	
-	
SDFLT4CMPL1	
-	
SDCPARM4	
-	See SDCPARM1
SDDATA4	
-	See SDDATA1
SDDATFIFO4	
-	
SDCDATA4	
-	
SDFLT4CMPH2	
-	
SDFLT4CMPHZ	
-	
SDFIFOCTL4	
-	
SDSYNC4	
-	
SDFLT4CMPL2	
-	
SDCOMP1CTL	
-	
SDCOMP1EVT2FLTCTL	
-	
SDCOMP1EVT2FLTCLKCTL	
-	
SDCOMP1EVT1FLTCTL	
-	

Table 32-9. SDFM Registers to Driverlib Functions (continued)

File	Driverlib Function
SDCOMP1EVT1FLTCLKCTL	
-	
SDCOMP1LOCK	
-	
SDCOMP2CTL	
-	
SDCOMP2EVT2FLTCTL	
-	
SDCOMP2EVT2FLTCLKCTL	
-	
SDCOMP2EVT1FLTCTL	
-	
SDCOMP2EVT1FLTCLKCTL	
-	
SDCOMP2LOCK	
-	
SDCOMP3CTL	
-	
SDCOMP3EVT2FLTCTL	
-	
SDCOMP3EVT2FLTCLKCTL	
-	
SDCOMP3EVT1FLTCTL	
-	
SDCOMP3EVT1FLTCLKCTL	
-	
SDCOMP3LOCK	
-	
SDCOMP4CTL	
-	
SDCOMP4EVT2FLTCTL	
-	
SDCOMP4EVT2FLTCLKCTL	
-	
SDCOMP4EVT1FLTCTL	
-	
SDCOMP4EVT1FLTCLKCTL	
-	
SDCOMP4LOCK	
-	

32.11.2 SDFM Examples

NOTE: These examples are located in the C2000Ware installation at the following location:
C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/sdfm

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

32.12 SDFM Registers

This Section describes the SDFM Registers.

32.12.1 SDFM Base Address Table

Table 32-10. SDFM Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
SDFM_REGS	SDFM1_BASE	0x7009_0000	YES	YES	YES	YES	YES	YES	-	YES
SDFM_REGS	SDFM2_BASE	0x7009_1000	YES	YES	YES	YES	YES	YES	-	YES
SDFM_REGS	SDFM3_BASE	0x7009_2000	YES	YES	YES	YES	YES	YES	-	YES
SDFM_REGS	SDFM4_BASE	0x7009_3000	YES	YES	YES	YES	YES	YES	-	YES

32.12.2 SDFM_REGS Registers

Table 32-11 lists the memory-mapped registers for the SDFM_REGS registers. All register offset addresses not listed in Table 32-11 should be considered as reserved locations and the register contents should not be modified.

Table 32-11. SDFM_REGS Registers

Offset	Acronym	Register Name	Protection
0h	SDIFLG	SD Interrupt Flag Register	
4h	SDIFLGCLR	SD Interrupt Flag Clear Register	
8h	SDCTL	SD Control Register	
Ch	SDMFILEN	SD Master Filter Enable	
Eh	SDSTATUS	SD Status Register	
10h	SDINTMODE	SD Interrupt Mode register	
20h	SDCTLPARM1	Control Parameter Register for Ch1	
22h	SDDFPARM1	Data Filter Parameter Register for Ch1	
24h	SDDPARM1	Data Parameter Register for Ch1	
26h	SDFLT1CMPH1	High-level Threshold Register for Ch1	
28h	SDFLT1CMPL1	Low-level Threshold Register for Ch1	
2Ah	SDCPARM1	Comparator Filter Parameter Register for Ch1	
2Ch	SDDATA1	Data Filter Data Register (16 or 32bit) for Ch1	
30h	SDDATFIFO1	Filter Data FIFO Output(32b) for Ch1	
34h	SDCDATA1	Comparator Filter Data Register (16b) for Ch1	
36h	SDFLT1CMPH2	Second high level threhold for CH1	
38h	SDFLT1CMPHZ	High-level (Z) Threshold Register for Ch1	
3Ah	SDFIFOCTL1	FIFO Control Register for Ch1	
3Ch	SDSYNC1	SD Filter Sync control for Ch1	
3Eh	SDFLT1CMPL2	Second low level threhold for CH1	
40h	SDCTLPARM2	Control Parameter Register for Ch2	
42h	SDDFPARM2	Data Filter Parameter Register for Ch2	
44h	SDDPARM2	Data Parameter Register for Ch2	
46h	SDFLT2CMPH1	High-level Threshold Register for Ch2	
48h	SDFLT2CMPL1	Low-level Threshold Register for Ch2	
4Ah	SDCPARM2	Comparator Filter Parameter Register for Ch2	
4Ch	SDDATA2	Data Filter Data Register (16 or 32bit) for Ch2	
50h	SDDATFIFO2	Filter Data FIFO Output(32b) for Ch2	
54h	SDCDATA2	Comparator Filter Data Register (16b) for Ch2	
56h	SDFLT2CMPH2	Second high level threhold for CH2	
58h	SDFLT2CMPHZ	High-level (Z) Threshold Register for Ch2	
5Ah	SDFIFOCTL2	FIFO Control Register for Ch2	
5Ch	SDSYNC2	SD Filter Sync control for Ch2	
5Eh	SDFLT2CMPL2	Second low level threhold for CH2	
60h	SDCTLPARM3	Control Parameter Register for Ch3	
62h	SDDFPARM3	Data Filter Parameter Register for Ch3	
64h	SDDPARM3	Data Parameter Register for Ch3	
66h	SDFLT3CMPH1	High-level Threshold Register for Ch3	
68h	SDFLT3CMPL1	Low-level Threshold Register for Ch3	
6Ah	SDCPARM3	Comparator Filter Parameter Register for Ch3	

Table 32-11. SDFM_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
6Ch	SDDATA3	Data Filter Data Register (16 or 32bit) for Ch3	
70h	SDDATFIFO3	Filter Data FIFO Output(32b) for Ch3	
74h	SDCDATA3	Comparator Filter Data Register (16b) for Ch3	
76h	SDFLT3CMPH2	Second high level threhold for CH3	
78h	SDFLT3CMPHZ	High-level (Z) Threshold Register for Ch3	
7Ah	SDFIFOCTL3	FIFO Control Register for Ch3	
7Ch	SDSYNC3	SD Filter Sync control for Ch3	
7Eh	SDFLT3CMPL2	Second low level threhold for CH3	
80h	SDCTLPARM4	Control Parameter Register for Ch4	
82h	SDDFPARM4	Data Filter Parameter Register for Ch4	
84h	SDDPARM4	Data Parameter Register for Ch4	
86h	SDFLT4CMPH1	High-level Threshold Register for Ch4	
88h	SDFLT4CMPL1	Low-level Threshold Register for Ch4	
8Ah	SDCPARM4	Comparator Filter Parameter Register for Ch4	
8Ch	SDDATA4	Data Filter Data Register (16 or 32bit) for Ch4	
90h	SDDATFIFO4	Filter Data FIFO Output(32b) for Ch4	
94h	SDCDATA4	Comparator Filter Data Register (16b) for Ch4	
96h	SDFLT4CMPH2	Second high level threhold for CH4	
98h	SDFLT4CMPHZ	High-level (Z) Threshold Register for Ch4	
9Ah	SDFIFOCTL4	FIFO Control Register for Ch4	
9Ch	SDSYNC4	SD Filter Sync control for Ch4	
9Eh	SDFLT4CMPL2	Second low level threhold for CH4	
C0h	SDCOMP1CTL	SD Comparator event filter1 Control Register	
C2h	SDCOMP1EVT2FLTCTL	COMPL/CEVT2 Digital filter1 Control Register	
C4h	SDCOMP1EVT2FLTCLKCTL	COMPL/CEVT2 Digital filter1 Clock Control Register	
C6h	SDCOMP1EVT1FLTCTL	COMPH/CEVT1 Digital filter1 Control Register	
C8h	SDCOMP1EVT1FLTCLKCTL	COMPH/CEVT1 Digital filter1 Clock Control Register	
CEh	SDCOMP1LOCK	SD compartor event filter1 Lock Register	
D0h	SDCOMP2CTL	SD Comparator event filter2 Control Register	
D2h	SDCOMP2EVT2FLTCTL	COMPL/CEVT2 Digital filter2 Control Register	
D4h	SDCOMP2EVT2FLTCLKCTL	COMPL/CEVT2 Digital filter2 Clock Control Register	
D6h	SDCOMP2EVT1FLTCTL	COMPH/CEVT1 Digital filter2 Control Register	
D8h	SDCOMP2EVT1FLTCLKCTL	COMPH/CEVT1 Digital filter2 Clock Control Register	
DEh	SDCOMP2LOCK	SD compartor event filter2 Lock Register	
E0h	SDCOMP3CTL	SD Comparator event filter3 Control Register	
E2h	SDCOMP3EVT2FLTCTL	COMPL/CEVT2 Digital filter3 Control Register	
E4h	SDCOMP3EVT2FLTCLKCTL	COMPL/CEVT2 Digital filter3 Clock Control Register	
E6h	SDCOMP3EVT1FLTCTL	COMPH/CEVT1 Digital filter3 Control Register	
E8h	SDCOMP3EVT1FLTCLKCTL	COMPH/CEVT1 Digital filter3 Clock Control Register	
EEh	SDCOMP3LOCK	SD compartor event filter3 Lock Register	
F0h	SDCOMP4CTL	SD Comparator event filter4 Control Register	
F2h	SDCOMP4EVT2FLTCTL	COMPL/CEVT2 Digital filter4 Control Register	
F4h	SDCOMP4EVT2FLTCLKCTL	COMPL/CEVT2 Digital filter4 Clock Control Register	
F6h	SDCOMP4EVT1FLTCTL	COMPH/CEVT1 Digital filter4 Control Register	
F8h	SDCOMP4EVT1FLTCLKCTL	COMPH/CEVT1 Digital filter4 Clock Control Register	

Table 32-11. SDFM_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
FEh	SDCOMP4LOCK	SD compartor event filter4 Lock Register	

Complex bit access types are encoded to fit into small table cells. Table 32-12 shows the codes that are used for access types in this section.

Table 32-12. SDFM_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

32.12.2.1 SDIFLG Register (Offset = 0h) [Reset = 00000000h]

SDIFLG is shown in Figure 32-15 and described in Table 32-13.

Return to the Summary Table.

SD Interrupt Flag Register

Figure 32-15. SDIFLG Register

31	30	29	28	27	26	25	24
MIF	RESERVED						
R-0h	R-0-0h						
23	22	21	20	19	18	17	16
SDFFINT4	SDFFINT3	SDFFINT2	SDFFINT1	SDFFOVF4	SDFFOVF3	SDFFOVF2	SDFFOVF1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
AF4	AF3	AF2	AF1	MF4	MF3	MF2	MF1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
FLT4_FLG_CEVT2	FLT4_FLG_CEVT1	FLT3_FLG_CEVT2	FLT3_FLG_CEVT1	FLT2_FLG_CEVT2	FLT2_FLG_CEVT1	FLT1_FLG_CEVT2	FLT1_FLG_CEVT1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 32-13. SDIFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MIF	R	0h	Set whenever any 'error' interrupt (MF1-4,IFL1-4,IFH1-4,SDFFOVF1-4) is active Reset type: SYSRSn
30-24	RESERVED	R-0	0h	Reserved
23	SDFFINT4	R	0h	SDFIFO data ready interrupt for Ch4 Reset type: SYSRSn
22	SDFFINT3	R	0h	SDFIFO data ready interrupt for Ch3 Reset type: SYSRSn
21	SDFFINT2	R	0h	SDFIFO data ready interrupt for Ch2 Reset type: SYSRSn
20	SDFFINT1	R	0h	SDFIFO data ready interrupt for Ch1 0: SDFIFO data ready interrupt has NOT occurred 1: SDFIFO data ready interrupt has occurred Reset type: SYSRSn
19	SDFFOVF4	R	0h	FIFO Overflow Flag for Ch4 Reset type: SYSRSn
18	SDFFOVF3	R	0h	FIFO Overflow Flag for Ch3 Reset type: SYSRSn
17	SDFFOVF2	R	0h	FIFO Overflow Flag for Ch2 Reset type: SYSRSn
16	SDFFOVF1	R	0h	FIFO Overflow Flag for Ch1 0 - FIFO has not overflowed 1 - FIFO overflowed. # words received in FIFO > FIFO depth (16), NEW word is lost Reset type: SYSRSn

Table 32-13. SDIFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	AF4	R	0h	Acknowledge flag for Filter 4 0: No new data available for Filter (in non-FIFO mode) 1: New data available for Filter (in non-FIFO mode) Reset type: SYSRSn
14	AF3	R	0h	Acknowledge flag for Filter 3 0: No new data available for Filter (in non-FIFO mode) 1: New data available for Filter (in non-FIFO mode) Reset type: SYSRSn
13	AF2	R	0h	Acknowledge flag for Filter 2 0: No new data available for Filter (in non-FIFO mode) 1: New data available for Filter (in non-FIFO mode) Reset type: SYSRSn
12	AF1	R	0h	Acknowledge flag for Filter 1 0: No new data available for Filter (in non-FIFO mode) 1: New data available for Filter (in non-FIFO mode) Reset type: SYSRSn
11	MF4	R	0h	Modulator Failure for Filter 4 0: Modulator is operating normally for Filter 1: Modulator failure for Filter Reset type: SYSRSn
10	MF3	R	0h	Modulator Failure for Filter 3 0: Modulator is operating normally for Filter 1: Modulator failure for Filter Reset type: SYSRSn
9	MF2	R	0h	Modulator Failure for Filter 2 0: Modulator is operating normally for Filter 1: Modulator failure for Filter Reset type: SYSRSn
8	MF1	R	0h	Modulator Failure for Filter 1 0: Modulator is operating normally for Filter 1: Modulator failure for Filter Reset type: SYSRSn
7	FLT4_FLG_CEVT2	R	0h	CEVT2 Interrupt flag for filter4 0: CEVT2 event has not occured 1: CEVT2 event has occurred Reset type: SYSRSn
6	FLT4_FLG_CEVT1	R	0h	CEVT1 Interrupt flag for filter4 0: CEVT1 event has not occured 1: CEVT1 event has occurred Reset type: SYSRSn
5	FLT3_FLG_CEVT2	R	0h	CEVT2 Interrupt flag for filter3 0: CEVT2 event has not occured 1: CEVT2 event has occurred Reset type: SYSRSn
4	FLT3_FLG_CEVT1	R	0h	CEVT1 Interrupt flag for filter3 0: CEVT1 event has not occured 1: CEVT1 event has occurred Reset type: SYSRSn
3	FLT2_FLG_CEVT2	R	0h	CEVT2 Interrupt flag for filter2 0: CEVT2 event has not occured 1: CEVT2 event has occurred Reset type: SYSRSn
2	FLT2_FLG_CEVT1	R	0h	CEVT1 Interrupt flag for filter2 0: CEVT1 event has not occured 1: CEVT1 event has occurred Reset type: SYSRSn

Table 32-13. SDIFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	FLT1_FLG_CEVT2	R	0h	CEVT2 Interrupt flag for filter1 0: CEVT2 event has not occured 1: CEVT2 event has occurred Reset type: SYSRSn
0	FLT1_FLG_CEVT1	R	0h	CEVT1 Interrupt flag for filter1 0: CEVT1 event has not occured 1: CEVT1 event has occurred Reset type: SYSRSn

32.12.2.2 SDIFLGCLR Register (Offset = 4h) [Reset = 00000000h]

SDIFLGCLR is shown in Figure 32-16 and described in Table 32-14.

Return to the Summary Table.

SD Module Interrupt Flag Clear Bits:
Writing a '1' will clear the respective flag bit in the SDIFLG register.
Writes of '0' are ignored.
Note: If user writes a '1' to clear a bit on the same cycle that the hardware is trying to set the bit to '1', then hardware has priority and the bit will not be cleared.

Figure 32-16. SDIFLGCLR Register

31	30	29	28	27	26	25	24
MIF	RESERVED						
R-0/W1S-0h	R-0-0h						
23	22	21	20	19	18	17	16
SDFFINT4	SDFFINT3	SDFFINT2	SDFFINT1	SDFFOVF4	SDFFOVF3	SDFFOVF2	SDFFOVF1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
AF4	AF3	AF2	AF1	MF4	MF3	MF2	MF1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
FLT4_FLG_CE VT2	FLT4_FLG_CE VT1	FLT3_FLG_CE VT2	FLT3_FLG_CE VT1	FLT2_FLG_CE VT2	FLT2_FLG_CE VT1	FLT1_FLG_CE VT2	FLT1_FLG_CE VT1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 32-14. SDIFLGCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MIF	R-0/W1S	0h	Flag-clear bit for SDFM Master Interrupt flag. Writing a 1 to clear MIF flag in SDIFLG register Writes of '0' are ignored. Note: If the MIF flag is cleared and other Interrupts are still pending, MIF will again be set to 1 on the following SysClk cycle, and the INT output will be reasserted (pulsed low) Reset type: SYSRSn
30-24	RESERVED	R-0	0h	Reserved
23	SDFFINT4	R-0/W1S	0h	SDFIFO data ready Interrupt flag-clear bit for Ch4 Reset type: SYSRSn
22	SDFFINT3	R-0/W1S	0h	SDFIFO data ready Interrupt flag-clear bit for Ch3 Reset type: SYSRSn
21	SDFFINT2	R-0/W1S	0h	SDFIFO data ready Interrupt flag-clear bit for Ch2 Reset type: SYSRSn
20	SDFFINT1	R-0/W1S	0h	SDFIFO data ready Interrupt flag-clear bit for Ch1 Reset type: SYSRSn
19	SDFFOVF4	R-0/W1S	0h	SDFIFO overflow clear Ch4 Reset type: SYSRSn
18	SDFFOVF3	R-0/W1S	0h	SDFIFO overflow clear Ch3 Reset type: SYSRSn
17	SDFFOVF2	R-0/W1S	0h	SDFIFO overflow clear Ch2 Reset type: SYSRSn
16	SDFFOVF1	R-0/W1S	0h	SDFIFO overflow clear Ch1 Reset type: SYSRSn

Table 32-14. SDIFLGCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	AF4	R-0/W1S	0h	Flag-clear bit for Acknowledge flag for Filter 4 Reset type: SYSRSn
14	AF3	R-0/W1S	0h	Flag Clear bit for AF3 Reset type: SYSRSn
13	AF2	R-0/W1S	0h	Flag Clear bit for AF2 Reset type: SYSRSn
12	AF1	R-0/W1S	0h	Flag Clear bit for AF1 Reset type: SYSRSn
11	MF4	R-0/W1S	0h	Flag Clear bit for MF4 Reset type: SYSRSn
10	MF3	R-0/W1S	0h	Flag Clear bit for MF3 Reset type: SYSRSn
9	MF2	R-0/W1S	0h	Flag Clear bit for MF2 Reset type: SYSRSn
8	MF1	R-0/W1S	0h	Flag Clear bit for MF1 Reset type: SYSRSn
7	FLT4_FLG_CEVT2	R-0/W1S	0h	Flag Clear bit for FLT4_FLG_CEVT2 Reset type: SYSRSn
6	FLT4_FLG_CEVT1	R-0/W1S	0h	Flag Clear bit for FLT4_FLG_CEVT1 Reset type: SYSRSn
5	FLT3_FLG_CEVT2	R-0/W1S	0h	Flag Clear bit for FLT3_FLG_CEVT2 Reset type: SYSRSn
4	FLT3_FLG_CEVT1	R-0/W1S	0h	Flag Clear bit for FLT3_FLG_CEVT1 Reset type: SYSRSn
3	FLT2_FLG_CEVT2	R-0/W1S	0h	Flag Clear bit for FLT2_FLG_CEVT2 Reset type: SYSRSn
2	FLT2_FLG_CEVT1	R-0/W1S	0h	Flag Clear bit for FLT2_FLG_CEVT1 Reset type: SYSRSn
1	FLT1_FLG_CEVT2	R-0/W1S	0h	Flag Clear bit for FLT1_FLG_CEVT2 Reset type: SYSRSn
0	FLT1_FLG_CEVT1	R-0/W1S	0h	Flag Clear bit for FLT1_FLG_CEVT1 Reset type: SYSRSn

32.12.2.3 SDCTL Register (Offset = 8h) [Reset = 0000h]

SDCTL is shown in Figure 32-17 and described in Table 32-15.

Return to the Summary Table.

SD Control Register

Figure 32-17. SDCTL Register

15	14	13	12	11	10	9	8
RESERVED	RESERVED	MIE	RESERVED				
R-0-0h	R-0-0h	R/W-0h	R-0-0h				
7	6	5	4	3	2	1	0
RESERVED				HZ4	HZ3	HZ2	HZ1
R-0-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 32-15. SDCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14	RESERVED	R-0	0h	Reserved
13	MIE	R/W	0h	Master SDy_ERR interrupt enable 0: SDy_ERR Interrupt and interrupt flags are disabled 1: SDy_ERR Interrupt and interrupt flags are enabled Reset type: SYSRSn
12-4	RESERVED	R-0	0h	Reserved
3	HZ4	R-0/W1S	0h	Flag Clear bit for HZ4 Reset type: SYSRSn
2	HZ3	R-0/W1S	0h	Flag Clear bit for HZ3 Reset type: SYSRSn
1	HZ2	R-0/W1S	0h	Flag Clear bit for HZ2 Reset type: SYSRSn
0	HZ1	R-0/W1S	0h	Flag Clear bit for HZ1 Reset type: SYSRSn

32.12.2.4 SDMFILEN Register (Offset = Ch) [Reset = 0000h]

SDMFILEN is shown in Figure 32-18 and described in Table 32-16.

Return to the Summary Table.

SD Master Filter Enable

Figure 32-18. SDMFILEN Register

15	14	13	12	11	10	9	8
RESERVED			RESERVED	MFE	RESERVED	RESERVED	RESERVED
R-0-0h			R-0-0h	R/W-0h	R-0-0h	R-0-0h	R-0-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED			RESERVED			
R-0-0h	R-0-0h			R-0-0h			

Table 32-16. SDMFILEN Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R-0	0h	Reserved
12	RESERVED	R-0	0h	Reserved
11	MFE	R/W	0h	Master Filter Enable 0: All the four data filter units of SDFM module are disabled. All FIFOs are cleared 1: Data filter units can be enabled if bit FEN is '1'. Reset type: SYSRSn
10	RESERVED	R-0	0h	Reserved
9	RESERVED	R-0	0h	Reserved
8-7	RESERVED	R-0	0h	Reserved
6-4	RESERVED	R-0	0h	Reserved
3-0	RESERVED	R-0	0h	Reserved

32.12.2.5 SDSTATUS Register (Offset = Eh) [Reset = 0000h]

SDSTATUS is shown in Figure 32-19 and described in Table 32-17.

Return to the Summary Table.

SD Status Register

Figure 32-19. SDSTATUS Register

15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED				HZ4	HZ3	HZ2	HZ1
R-0-0h				R-0h	R-0h	R-0h	R-0h

Table 32-17. SDSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7-4	RESERVED	R-0	0h	Reserved
3	HZ4	R	0h	High-level Threshold crossing (Z) flag Ch4 Primarily intended for detecting 'zero'-crossing events. Unlike the primary comparator IFHx flag, it does not have the ability to generate an interrupt. 0: Comparator filter output < SDCMPHZ4.HLTZ 1: Comparator filter output >= SDCMPHZ4.HLTZ Reset type: SYSRSn
2	HZ3	R	0h	High-level Threshold crossing (Z) flag Ch3 Primarily intended for detecting 'zero'-crossing events. Unlike the primary comparator IFHx flag, it does not have the ability to generate an interrupt. 0: Comparator filter output < SDCMPHZ3.HLTZ 1: Comparator filter output >= SDCMPHZ3.HLTZ Reset type: SYSRSn
1	HZ2	R	0h	High-level Threshold crossing (Z) flag Ch2 Primarily intended for detecting 'zero'-crossing events. Unlike the primary comparator IFHx flag, it does not have the ability to generate an interrupt. 0: Comparator filter output < SDCMPHZ2.HLTZ 1: Comparator filter output >= SDCMPHZ2.HLTZ Reset type: SYSRSn
0	HZ1	R	0h	High-level Threshold crossing (Z) flag Ch1 Primarily intended for detecting 'zero'-crossing events. Unlike the primary comparator IFHx flag, it does not have the ability to generate an interrupt. 0: Comparator filter output < SDCMPHZ1.HLTZ 1: Comparator filter output >= SDCMPHZ1.HLTZ Reset type: SYSRSn

32.12.2.6 SDINTMODE Register (Offset = 10h) [Reset = 0000h]

SDINTMODE is shown in Figure 32-20 and described in Table 32-18.

Return to the Summary Table.

SD Interrupt Mode register

Figure 32-20. SDINTMODE Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							SDINTMODESEL
R-0h							R/W-0h

Table 32-18. SDINTMODE Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R	0h	Reserved
0	SDINTMODESEL	R/W	0h	CompxH/L events interrupt mode select 0 CompxH/L events are treated as edge signals, rise-edge detect will be done to qualify the event for interrupt generation 1 CompxH/L events are treated as level signals. Rise-edge detect will not be performed and interrupt will be re-asserted if the event remains asserted. Reset type: SYSRSn

32.12.2.7 SDCTLPARM1 Register (Offset = 20h) [Reset = 0000h]

SDCTLPARM1 is shown in Figure 32-21 and described in Table 32-19.

Return to the Summary Table.

Control Parameter Register for Ch1

Figure 32-21. SDCTLPARM1 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	SDDATASYNC	RESERVED	SDCLKSYNC	SDCLKSEL	RESERVED	MOD	
R-0-0h	R/W-0h	R-0-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

Table 32-19. SDCTLPARM1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R-0	0h	Reserved
7	RESERVED	R-0	0h	Reserved
6	SDDATASYNC	R/W	0h	0: SD Data is not passed through a synchronizer. 1: SD Data is passed through a synchronizer. Reset type: SYSRSn
5	RESERVED	R-0	0h	Reserved
4	SDCLKSYNC	R/W	0h	0: SD Clock is not passed through a synchronizer. 1: SD Clock is passed through a synchronizer. Reset type: SYSRSn
3	SDCLKSEL	R/W	0h	SD1 Clock source select. 0: Clock source to SDFM filter is its channel clock. 1: Clock source to SDFM filter is SD1 filter clock. Reset type: SYSRSn
2	RESERVED	R/W	0h	Reserved
1-0	MOD	R/W	0h	Modulator clock modes 0: Mode 0: Modulator clock running at 1x data rate 1: Reserved 2: Reserved 3: Reserved Reset type: SYSRSn

32.12.2.8 SDDFPARM1 Register (Offset = 22h) [Reset = 0000h]

SDDFPARM1 is shown in Figure 32-22 and described in Table 32-20.

Return to the Summary Table.

Data Filter Parameter Register for Ch1

Figure 32-22. SDDFPARM1 Register

15	14	13	12	11	10	9	8
RESERVED			SDSYNCEN	SST		AE	FEN
R-0h			R/W-0h	R/W-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DOSR							
R/W-0h							

Table 32-20. SDDFPARM1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	SDSYNCEN	R/W	0h	PWM synchronization (SDSYNC) of data filter 0: PWM synchronization of data filter is disabled 1: PWM synchronization of data filter is enabled Note: SDSYNCx.SYNCSEL bits define which PWM signal is used to synchronize PWMs Reset type: SYSRSn
11-10	SST	R/W	0h	Data filter structure 00: Data filter runs with a Sincfast structure 01: Data filter runs with a Sinc1 structure 10: Data filter runs with a Sinc2 structure 11: Data filter runs with a Sinc3 structure Reset type: SYSRSn
9	AE	R/W	0h	Data filter Acknowledge Enable 0: Acknowledge flag is disabled for the particular filter 1: Acknowledge flag is enabled for the particular filter Reset type: SYSRSn
8	FEN	R/W	0h	Filter Enable 0: The data filter is disabled and no data is produced 1: The data filter is enabled and data are produced in the data filter Note: When filter is disabled, DOSR counter held in reset, filter data erased. Also resets FIFO pointers and clears the FIFO Reset type: SYSRSn
7-0	DOSR	R/W	0h	Data filter Oversampling ratio The actual oversampling ratio of data filter is DOSR + 1 These bits set the oversampling ratio of the data filter. 0x0FF represents an oversampling ratio of 256. Reset type: SYSRSn

32.12.2.9 SDDPARM1 Register (Offset = 24h) [Reset = 0000h]

SDDPARM1 is shown in Figure 32-23 and described in Table 32-21.

Return to the Summary Table.

Data Parameter Register for Ch1

Figure 32-23. SDDPARM1 Register

15	14	13	12	11	10	9	8
SH					DR	RESERVED	
R/W-0h					R/W-0h	R-0-0h	
7	6	5	4	3	2	1	0
RESERVED							
R-0-0h							

Table 32-21. SDDPARM1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	SH	R/W	0h	Shift Control These bits indicate by how many bits the 16-bit window is shifted up when 16-bit data representation is chosen. Reset type: SYSRSn
10	DR	R/W	0h	Data filter Data representation 0: Data stored in 16b 2's complement 1: Data stored in 32b 2's complement Reset type: SYSRSn
9-0	RESERVED	R-0	0h	Reserved

32.12.2.10 SDFLT1CMPH1 Register (Offset = 26h) [Reset = 7FFFh]

SDFLT1CMPH1 is shown in Figure 32-24 and described in Table 32-22.

Return to the Summary Table.

High-level Threshold Register for Ch1

Figure 32-24. SDFLT1CMPH1 Register

15	14	13	12	11	10	9	8
RESERVED	HLT						
R-0-0h	R/W-7FFFh						
7	6	5	4	3	2	1	0
HLT							
R/W-7FFFh							

Table 32-22. SDFLT1CMPH1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	HLT	R/W	7FFFh	Unsigned high-level threshold for the comparator filter output. Reset type: SYSRSn

32.12.2.11 SDFLT1CMPL1 Register (Offset = 28h) [Reset = 0000h]

SDFLT1CMPL1 is shown in Figure 32-25 and described in Table 32-23.

Return to the Summary Table.

Low-level Threshold Register for Ch1

Figure 32-25. SDFLT1CMPL1 Register

15	14	13	12	11	10	9	8
RESERVED	LLT						
R-0-0h	R/W-0h						
7	6	5	4	3	2	1	0
LLT							
R/W-0h							

Table 32-23. SDFLT1CMPL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	LLT	R/W	0h	Unsigned low-level threshold for the comparator filter output. Reset type: SYSRSn

32.12.2.12 SDCPARM1 Register (Offset = 2Ah) [Reset = 2000h]

SDCPARM1 is shown in Figure 32-26 and described in Table 32-24.

Return to the Summary Table.

Comparator Filter Parameter Register for Ch1

Figure 32-26. SDCPARM1 Register

15	14	13	12	11	10	9	8
CEVT2SEL		CEN	CEVT1SEL		HZEN	MFIE	CS1_CS0
R/W-0h		R/W-1h	R/W-0h		R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CS1_CS0	EN_CEVT2	EN_CEVT1	COSR				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				

Table 32-24. SDCPARM1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	CEVT2SEL	R/W	0h	Comparator Event2 Select 00: COMPL1 01: COMPL1 OR COMPH1 10: COMPL2 11: COMPL2 OR COMPH2 Reset type: SYSRSn
13	CEN	R/W	1h	Comparator Filter enable 0: Disable comparator filter 1: Enable comparator filter Reset type: SYSRSn
12-11	CEVT1SEL	R/W	0h	Comparator Event1 Select 00: COMPH1 01: COMPL1 OR COMPH1 10: COMPH2 11: COMPL2 OR COMPH2 Reset type: SYSRSn
10	HZEN	R/W	0h	High level (Z) Threshold crossing output enable 0: Disable Higher level Threshold (Z) crossing 1: Enable Higher level Threhold (Z) crossing Reset type: SYSRSn
9	MFIE	R/W	0h	Modulator Failure Interrupt Enable 0: Disable modulator failure interrupt and its flag 1: Enable modulator failure interrupt and its flag Reset type: SYSRSn
8-7	CS1_CS0	R/W	0h	Comparator filter structure 00: Comparator filter runs with a sincfast structure 01: Comparator filter runs with a Sinc1 structure 10: Comparator filter runs with a Sinc2 structure 11: Comparator filter runs with a Sinc3 structure Reset type: SYSRSn
6	EN_CEVT2	R/W	0h	CEVT2 interrupt enable 0: Disable CEVT2 interrupt 1: Enable CEVT2 interrupt Reset type: SYSRSn
5	EN_CEVT1	R/W	0h	CEVT1 interrupt enable 0: Disable CEVT1 interrupt 1: Enable CEVT1 interrupt Reset type: SYSRSn

Table 32-24. SDCPARM1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	COSR	R/W	0h	Comparator Oversampling ratio. The actual rate is COSR + 1. These bits set the oversampling ratio of the filter. 0x1F represents an oversampling ratio of 32 Reset type: SYSRSn

32.12.2.13 SDDATA1 Register (Offset = 2Ch) [Reset = 00000000h]

SDDATA1 is shown in Figure 32-27 and described in Table 32-25.

Return to the Summary Table.

Data Filter Data Register (16 or 32bit) for Ch1

Figure 32-27. SDDATA1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA32HI																DATA16															
R-0h																R-0h															

Table 32-25. SDDATA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DATA32HI	R	0h	Hi-order 16b in 32b mode, 16-bit Data in 16b mode Reset type: SYSRSn
15-0	DATA16	R	0h	Lo-order 16b in 32b mode Reset type: SYSRSn

32.12.2.14 SDDATFIFO1 Register (Offset = 30h) [Reset = 00000000h]

SDDATFIFO1 is shown in Figure 32-28 and described in Table 32-26.

Return to the Summary Table.

Filter Data FIFO Output(32b) for Ch1

Figure 32-28. SDDATFIFO1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA32HI																DATA16															
R-0h																R-0h															

Table 32-26. SDDATFIFO1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DATA32HI	R	0h	Hi-order 16b in 32b mode, 16-bit Data in 16b mode Reset type: SYSRSn
15-0	DATA16	R	0h	Lo-order 16b in 32b mode Reset type: SYSRSn

32.12.2.15 SDCDATA1 Register (Offset = 34h) [Reset = 0000h]

SDCDATA1 is shown in Figure 32-29 and described in Table 32-27.

Return to the Summary Table.

Comparator Filter Data Register (16b) for Ch1

Figure 32-29. SDCDATA1 Register

15	14	13	12	11	10	9	8
DATA16							
R-0h							
7	6	5	4	3	2	1	0
DATA16							
R-0h							

Table 32-27. SDCDATA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	DATA16	R	0h	Comparator Data output - 16b only Reset type: SYSRSn

32.12.2.16 SDFLT1CMPH2 Register (Offset = 36h) [Reset = 7FFFh]

SDFLT1CMPH2 is shown in Figure 32-30 and described in Table 32-28.

Return to the Summary Table.

Second high level threhold for CH1

Figure 32-30. SDFLT1CMPH2 Register

15	14	13	12	11	10	9	8
RESERVED	HLT2						
R-0-0h	R/W-7FFFh						
7	6	5	4	3	2	1	0
HLT2							
R/W-7FFFh							

Table 32-28. SDFLT1CMPH2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	HLT2	R/W	7FFFh	Second Unsigned high-level threshold for the comparator filter output. Reset type: SYSRSn

32.12.2.17 SDFLT1CMPHZ Register (Offset = 38h) [Reset = 0000h]

SDFLT1CMPHZ is shown in Figure 32-31 and described in Table 32-29.

Return to the Summary Table.

High-level (Z) Threshold Register for Ch1

Figure 32-31. SDFLT1CMPHZ Register

15	14	13	12	11	10	9	8
RESERVED	HLTZ						
R-0-0h	R/W-0h						
7	6	5	4	3	2	1	0
HLTZ							
R/W-0h							

Table 32-29. SDFLT1CMPHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	HLTZ	R/W	0h	Unsigned High-level threshold (Z) for the comparator filter output Primarily intended for detecting 'zero'-crossing events. Unlike the primary comparator SDCMPHx, it does not have the ability to generate an interrupt. Reset type: SYSRSn

32.12.2.18 SDFIFOCTL1 Register (Offset = 3Ah) [Reset = 0000h]

SDFIFOCTL1 is shown in Figure 32-32 and described in Table 32-30.

Return to the Summary Table.

FIFO Control Register for Ch1

Figure 32-32. SDFIFOCTL1 Register

15	14	13	12	11	10	9	8
OVFIEN	DRINTSEL	FFEN	FFIEN	RESERVED	SDFFST		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0-0h	R-0h		
7	6	5	4	3	2	1	0
SDFFST		RESERVED	SDFFIL				
R-0h		R-0-0h	R/W-0h				

Table 32-30. SDFIFOCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	OVFIEN	R/W	0h	SDFIFO Overflow interrupt enable 0: SDFIFO Overflow condition will not generate an interrupt 1: SDFIFO overflow condition generates an interrupt on SDy_ERR Reset type: SYSRSn
14	DRINTSEL	R/W	0h	Data-Ready Interrupt (DRINT) source select 0 = AF1 (Select non-FIFO data-ready interrupt) 1 = SDFFINT1 (Select FIFO data-ready interrupt) Reset type: SYSRSn
13	FFEN	R/W	0h	SDFIFO Enable 0: Disable FIFO operation 1: Enable FIFO operation Note: When FIFO is disabled, FIFO contents are cleared Reset type: SYSRSn
12	FFIEN	R/W	0h	SDFIFO data ready Interrupt Enable Reset type: SYSRSn
11	RESERVED	R-0	0h	Reserved
10-6	SDFFST	R	0h	SDFIFO Status 00000 FIFO empty 00001 FIFO has 1 word . . . . 10000 FIFO has 16 words Reset type: SYSRSn
5	RESERVED	R-0	0h	Reserved
4-0	SDFFIL	R/W	0h	SDFIFO interrupt level bits The FIFO will generate an interrupt when the FIFO status (SDFFST) >= FIFO level (SDFFIL) Reset type: SYSRSn

32.12.2.19 SDSYNC1 Register (Offset = 3Ch) [Reset = 043Fh]

SDSYNC1 is shown in Figure 32-33 and described in Table 32-31.

Return to the Summary Table.

SD Filter Sync control for Ch1

Figure 32-33. SDSYNC1 Register

15	14	13	12	11	10	9	8
RESERVED					WTSCLREN	FFSYNCCLREN	WTSYNCLR
R-0-0h					R/W-1h	R/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
WTSYNFLG	WTSYNCEN	SYNCSEL					
R-0h	R/W-0h	R/W-3Fh					

Table 32-31. SDSYNC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R-0	0h	Reserved
10	WTSCLREN	R/W	1h	WTSYNFLG Clear-on-FIFOINT Enable 0: WTSYNFLG can only be cleared manually (using WTSYNCLR bit) 1: WTSYNFLG is cleared automatically on SDFFINT Reset type: SYSRSn
9	FFSYNCCLREN	R/W	0h	FIFO Clear-on-SDSYNC Enable 0: SDFIFO is not automaticaly cleared upon receiving SDSYNC 1: SDFIFO is automaticaly cleared upon receiving SDSYNC Reset type: SYSRSn
8	WTSYNCLR	R-0/W	0h	Wait-for-Sync Flag Clear (always reads 0) 0: Write of 0 has no affect 1: Write of 1 clears WTSYNFLG Reset type: SYSRSn
7	WTSYNFLG	R	0h	Wait-for-Sync Flag 0: SDSYNC event has not occurred 1: SDSYNC event occurred. Reset type: SYSRSn
6	WTSYNCEN	R/W	0h	Wait-for-Sync Enable 0: Incoming Data written to SDFIFO on every Data-Ready (DR) Event 1: Incoming Data written to SDFIFO on DR event only after SDSYNC event occurs Reset type: SYSRSn
5-0	SYNCSEL	R/W	3Fh	Defines source for the SDSYNC Input on this channel Refer SDSYNCx.SYNCSEL table Reset type: SYSRSn

32.12.2.20 SDFLT1CMPL2 Register (Offset = 3Eh) [Reset = 0000h]

SDFLT1CMPL2 is shown in Figure 32-34 and described in Table 32-32.

Return to the Summary Table.

Second low level threhold for CH1

Figure 32-34. SDFLT1CMPL2 Register

15	14	13	12	11	10	9	8
RESERVED	LLT2						
R-0-0h	R/W-0h						
7	6	5	4	3	2	1	0
LLT2							
R/W-0h							

Table 32-32. SDFLT1CMPL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	LLT2	R/W	0h	Second Unsigned low-level threshold for the comparator filter output. Reset type: SYSRSn

32.12.2.21 SDCTLPARM2 Register (Offset = 40h) [Reset = 0000h]

SDCTLPARM2 is shown in Figure 32-35 and described in Table 32-33.

Return to the Summary Table.

Control Parameter Register for Ch2

Figure 32-35. SDCTLPARM2 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	SDDATASYNC	RESERVED	SDCLKSYNC	SDCLKSEL	RESERVED	MOD	
R-0-0h	R/W-0h	R-0-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

Table 32-33. SDCTLPARM2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	RESERVED	R-0	0h	Reserved
6	SDDATASYNC	R/W	0h	0: SD Data is not passed through a synchronizer. 1: SD Data is passed through a synchronizer. Reset type: SYSRSn
5	RESERVED	R-0	0h	Reserved
4	SDCLKSYNC	R/W	0h	0: SD Clock is not passed through a synchronizer. 1: SD Clock is passed through a synchronizer. Reset type: SYSRSn
3	SDCLKSEL	R/W	0h	SD2 Clock source select. 0: Clock source to SDFM filter is its channel clock. 1: Clock source to SDFM filter is SD1 filter clock. Reset type: SYSRSn
2	RESERVED	R/W	0h	Reserved
1-0	MOD	R/W	0h	Modulator clock modes 0: Mode 0: Modulator clock running at 1x data rate 1: Reserved 2: Reserved 3: Reserved Reset type: SYSRSn

32.12.2.22 SDDFPARM2 Register (Offset = 42h) [Reset = 0000h]

SDDFPARM2 is shown in Figure 32-36 and described in Table 32-34.

Return to the Summary Table.

Data Filter Parameter Register for Ch2

Figure 32-36. SDDFPARM2 Register

15	14	13	12	11	10	9	8
RESERVED			SDSYNCEN	SST		AE	FEN
R-0h			R/W-0h	R/W-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DOSR							
R/W-0h							

Table 32-34. SDDFPARM2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	SDSYNCEN	R/W	0h	PWM synchronization (SDSYNC) of data filter 0: PWM synchronization of data filter is disabled 1: PWM synchronization of data filter is enabled Note: SDSYNCx.SYNCSEL bits define which PWM signal is used to synchronize PWMs Reset type: SYSRSn
11-10	SST	R/W	0h	Data filter structure 00: Data filter runs with a Sincfast structure 01: Data filter runs with a Sinc1 structure 10: Data filter runs with a Sinc2 structure 11: Data filter runs with a Sinc3 structure Reset type: SYSRSn
9	AE	R/W	0h	Data filter Acknowledge Enable 0: Acknowledge flag is disabled for the particular filter 1: Acknowledge flag is enabled for the particular filter Reset type: SYSRSn
8	FEN	R/W	0h	Filter Enable 0: The data filter is disabled and no data is produced 1: The data filter is enabled and data are produced in the data filter Note: When filter is disabled, DOSR counter held in reset, filter data erased. Also resets FIFO pointers and clears the FIFO Reset type: SYSRSn
7-0	DOSR	R/W	0h	Data filter Oversampling ratio The actual oversampling ratio of data filter is DOSR + 1 These bits set the oversampling ratio of the data filter. 0x0FF represents an oversampling ratio of 256. Reset type: SYSRSn

32.12.2.23 SDDPARM2 Register (Offset = 44h) [Reset = 0000h]

SDDPARM2 is shown in Figure 32-37 and described in Table 32-35.

Return to the Summary Table.

Data Parameter Register for Ch2

Figure 32-37. SDDPARM2 Register

15	14	13	12	11	10	9	8
SH					DR	RESERVED	
R/W-0h					R/W-0h	R-0-0h	
7	6	5	4	3	2	1	0
RESERVED							
R-0-0h							

Table 32-35. SDDPARM2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	SH	R/W	0h	Shift Control These bits indicate by how many bits the 16-bit window is shifted up when 16-bit data representation is chosen. Reset type: SYSRSn
10	DR	R/W	0h	Data filter Data representation 0: Data stored in 16b 2's complement 1: Data stored in 32b 2's complement Reset type: SYSRSn
9-0	RESERVED	R-0	0h	Reserved

32.12.2.24 SDFLT2CMPH1 Register (Offset = 46h) [Reset = 7FFFh]

SDFLT2CMPH1 is shown in Figure 32-38 and described in Table 32-36.

Return to the Summary Table.

High-level Threshold Register for Ch2

Figure 32-38. SDFLT2CMPH1 Register

15	14	13	12	11	10	9	8
RESERVED	HLT						
R-0-0h	R/W-7FFFh						
7	6	5	4	3	2	1	0
HLT							
R/W-7FFFh							

Table 32-36. SDFLT2CMPH1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	HLT	R/W	7FFFh	Unsigned high-level threshold for the comparator filter output. Reset type: SYSRSn

32.12.2.25 SDFLT2CMPL1 Register (Offset = 48h) [Reset = 0000h]

SDFLT2CMPL1 is shown in Figure 32-39 and described in Table 32-37.

Return to the Summary Table.

Low-level Threshold Register for Ch2

Figure 32-39. SDFLT2CMPL1 Register

15	14	13	12	11	10	9	8
RESERVED	LLT						
R-0-0h	R/W-0h						
7	6	5	4	3	2	1	0
LLT							
R/W-0h							

Table 32-37. SDFLT2CMPL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	LLT	R/W	0h	Unsigned low-level threshold for the comparator filter output. Reset type: SYSRSn

32.12.2.26 SDCPARM2 Register (Offset = 4Ah) [Reset = 2000h]

SDCPARM2 is shown in Figure 32-40 and described in Table 32-38.

Return to the Summary Table.

Comparator Filter Parameter Register for Ch2

Figure 32-40. SDCPARM2 Register

15	14	13	12	11	10	9	8
CEVT2SEL		CEN	CEVT1SEL		HZEN	MFIE	CS1_CS0
R/W-0h		R/W-1h	R/W-0h		R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CS1_CS0	EN_CEVT2	EN_CEVT1	COSR				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				

Table 32-38. SDCPARM2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	CEVT2SEL	R/W	0h	Comparator Event2 Select 00: COMPL1 01: COMPL1 OR COMPH1 10: COMPL2 11: COMPL2 OR COMPH2 Reset type: SYSRSn
13	CEN	R/W	1h	Comparator Filter enable 0: Disable comparator filter 1: Enable comparator filter Reset type: SYSRSn
12-11	CEVT1SEL	R/W	0h	Comparator Event1 Select 00: COMPH1 01: COMPL1 OR COMPH1 10: COMPH2 11: COMPL2 OR COMPH2 Reset type: SYSRSn
10	HZEN	R/W	0h	High level (Z) Threshold crossing output enable 0: Disable Higher level Threshold (Z) crossing 1: Enable Higher level Threhold (Z) crossing Reset type: SYSRSn
9	MFIE	R/W	0h	Modulator Failure Interrupt Enable 0: Disable modulator failure interrupt and its flag 1: Enable modulator failure interrupt and its flag Reset type: SYSRSn
8-7	CS1_CS0	R/W	0h	Comparator filter structure 00: Comparator filter runs with a sincfast structure 01: Comparator filter runs with a Sinc1 structure 10: Comparator filter runs with a Sinc2 structure 11: Comparator filter runs with a Sinc3 structure Reset type: SYSRSn
6	EN_CEVT2	R/W	0h	CEVT2 interrupt enable 0: Disable CEVT2 interrupt 1: Enable CEVT2 interrupt Reset type: SYSRSn
5	EN_CEVT1	R/W	0h	CEVT1 interrupt enable 0: Disable CEVT1 interrupt 1: Enable CEVT1 interrupt Reset type: SYSRSn

Table 32-38. SDCPARM2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	COSR	R/W	0h	Comparator Oversampling ratio. The actual rate is COSR + 1. These bits set the oversampling ratio of the filter. 0x1F represents an oversampling ratio of 32 Reset type: SYSRSn

32.12.2.27 SDDATA2 Register (Offset = 4Ch) [Reset = 00000000h]

SDDATA2 is shown in Figure 32-41 and described in Table 32-39.

Return to the Summary Table.

Data Filter Data Register (16 or 32bit) for Ch2

Figure 32-41. SDDATA2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA32HI																DATA16															
R-0h																R-0h															

Table 32-39. SDDATA2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DATA32HI	R	0h	Hi-order 16b in 32b mode, 16-bit Data in 16b mode Reset type: SYSRSn
15-0	DATA16	R	0h	Lo-order 16b in 32b mode Reset type: SYSRSn

32.12.2.28 SDDATFIFO2 Register (Offset = 50h) [Reset = 00000000h]

SDDATFIFO2 is shown in Figure 32-42 and described in Table 32-40.

Return to the Summary Table.

Filter Data FIFO Output(32b) for Ch2

Figure 32-42. SDDATFIFO2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA32HI																DATA16															
R-0h																R-0h															

Table 32-40. SDDATFIFO2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DATA32HI	R	0h	Hi-order 16b in 32b mode, 16-bit Data in 16b mode Reset type: SYSRSn
15-0	DATA16	R	0h	Lo-order 16b in 32b mode Reset type: SYSRSn

32.12.2.29 SDCDATA2 Register (Offset = 54h) [Reset = 0000h]

SDCDATA2 is shown in Figure 32-43 and described in Table 32-41.

Return to the Summary Table.

Comparator Filter Data Register (16b) for Ch2

Figure 32-43. SDCDATA2 Register

15	14	13	12	11	10	9	8
DATA16							
R-0h							
7	6	5	4	3	2	1	0
DATA16							
R-0h							

Table 32-41. SDCDATA2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	DATA16	R	0h	Comparator Data output - 16b only Reset type: SYSRSn

32.12.2.30 SDFLT2CMPH2 Register (Offset = 56h) [Reset = 7FFFh]

SDFLT2CMPH2 is shown in Figure 32-44 and described in Table 32-42.

Return to the Summary Table.

Second high level threhold for CH2

Figure 32-44. SDFLT2CMPH2 Register

15	14	13	12	11	10	9	8
RESERVED	HLT2						
R-0-0h	R/W-7FFFh						
7	6	5	4	3	2	1	0
HLT2							
R/W-7FFFh							

Table 32-42. SDFLT2CMPH2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	HLT2	R/W	7FFFh	Second Unsigned high-level threshold for the comparator filter output. Reset type: SYSRSn

32.12.2.31 SDFLT2CMPHZ Register (Offset = 58h) [Reset = 0000h]

SDFLT2CMPHZ is shown in Figure 32-45 and described in Table 32-43.

Return to the Summary Table.

High-level (Z) Threshold Register for Ch2

Figure 32-45. SDFLT2CMPHZ Register

15	14	13	12	11	10	9	8
RESERVED	HLTZ						
R-0-0h	R/W-0h						
7	6	5	4	3	2	1	0
HLTZ							
R/W-0h							

Table 32-43. SDFLT2CMPHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	HLTZ	R/W	0h	Unsigned High-level threshold (Z) for the comparator filter output Primarily intended for detecting 'zero'-crossing events. Unlike the primary comparator SDCMPHx, it does not have the ability to generate an interrupt. Reset type: SYSRSn

32.12.2.32 SDFIFOCTL2 Register (Offset = 5Ah) [Reset = 0000h]

SDFIFOCTL2 is shown in Figure 32-46 and described in Table 32-44.

Return to the Summary Table.

FIFO Control Register for Ch2

Figure 32-46. SDFIFOCTL2 Register

15	14	13	12	11	10	9	8
OVFIEN	DRINTSEL	FFEN	FFIEN	RESERVED	SDFFST		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0-0h	R-0h		
7	6	5	4	3	2	1	0
SDFFST		RESERVED	SDFFIL				
R-0h		R-0-0h	R/W-0h				

Table 32-44. SDFIFOCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	OVFIEN	R/W	0h	SDFIFO Overflow interrupt enable 0: SDFIFO Overflow condition will not generate an interrupt 1: SDFIFO overflow condition generates an interrupt on SDy_ERR Reset type: SYSRSn
14	DRINTSEL	R/W	0h	Data-Ready Interrupt (DRINT) source select 0 = AF2 (Select non-FIFO data-ready interrupt) 1 = SDFFINT2 (Select FIFO data-ready interrupt) Reset type: SYSRSn
13	FFEN	R/W	0h	SDFIFO Enable 0: Disable FIFO operation 1: Enable FIFO operation Note: When FIFO is disabled, FIFO contents are cleared Reset type: SYSRSn
12	FFIEN	R/W	0h	SDFIFO data ready Interrupt Enable Reset type: SYSRSn
11	RESERVED	R-0	0h	Reserved
10-6	SDFFST	R	0h	SDFIFO Status 00000 FIFO empty 00001 FIFO has 1 word . . . . 10000 FIFO has 16 words Reset type: SYSRSn
5	RESERVED	R-0	0h	Reserved
4-0	SDFFIL	R/W	0h	SDFIFO interrupt level bits The FIFO will generate an interrupt when the FIFO status (SDFFST) >= FIFO level (SDFFIL) Reset type: SYSRSn

32.12.2.33 SDSYNC2 Register (Offset = 5Ch) [Reset = 043Fh]

SDSYNC2 is shown in Figure 32-47 and described in Table 32-45.

Return to the Summary Table.

SD Filter Sync control for Ch2

Figure 32-47. SDSYNC2 Register

15	14	13	12	11	10	9	8
RESERVED					WTSCLREN	FFSYNCCLREN	WTSYNCLR
R-0-0h					R/W-1h	R/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
WTSYNFLG	WTSYNCEN	SYNCSEL					
R-0h	R/W-0h	R/W-3Fh					

Table 32-45. SDSYNC2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R-0	0h	Reserved
10	WTSCLREN	R/W	1h	WTSYNFLG Clear-on-FIFOINT Enable 0: WTSYNFLG can only be cleared manually (using WTSYNCLR bit) 1: WTSYNFLG is cleared automatically on SDFFINT Reset type: SYSRSn
9	FFSYNCCLREN	R/W	0h	FIFO Clear-on-SDSYNC Enable 0: SDFIFO is not automaticaly cleared upon receiving SDSYNC 1: SDFIFO is automaticaly cleared upon receiving SDSYNC Reset type: SYSRSn
8	WTSYNCLR	R-0/W	0h	Wait-for-Sync Flag Clear (always reads 0) 0: Write of 0 has no affect 1: Write of 1 clears WTSYNFLG Reset type: SYSRSn
7	WTSYNFLG	R	0h	Wait-for-Sync Flag 0: SDSYNC event has not occurred 1: SDSYNC event occurred. Reset type: SYSRSn
6	WTSYNCEN	R/W	0h	Wait-for-Sync Enable 0: Incoming Data written to SDFIFO on every Data-Ready (DR) Event 1: Incoming Data written to SDFIFO on DR event only after SDSYNC event occurs Reset type: SYSRSn
5-0	SYNCSEL	R/W	3Fh	Defines source for the SDSYNC Input on this channel Refer SDSYNCx.SYNCSEL table Reset type: SYSRSn

32.12.2.34 SDFLT2CMPL2 Register (Offset = 5Eh) [Reset = 0000h]

SDFLT2CMPL2 is shown in Figure 32-48 and described in Table 32-46.

Return to the Summary Table.

Second low level threhold for CH2

Figure 32-48. SDFLT2CMPL2 Register

15	14	13	12	11	10	9	8
RESERVED	LLT2						
R-0-0h	R/W-0h						
7	6	5	4	3	2	1	0
LLT2							
R/W-0h							

Table 32-46. SDFLT2CMPL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	LLT2	R/W	0h	Second Unsigned low-level threshold for the comparator filter output. Reset type: SYSRSn

32.12.2.35 SDCTLPARM3 Register (Offset = 60h) [Reset = 0000h]

SDCTLPARM3 is shown in Figure 32-49 and described in Table 32-47.

Return to the Summary Table.

Control Parameter Register for Ch3

Figure 32-49. SDCTLPARM3 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	SDDATASYNC	RESERVED	SDCLKSYNC	SDCLKSEL	RESERVED	MOD	
R-0-0h	R/W-0h	R-0-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

Table 32-47. SDCTLPARM3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	RESERVED	R-0	0h	Reserved
6	SDDATASYNC	R/W	0h	0: SD Data is not passed through a synchronizer. 1: SD Data is passed through a synchronizer. Reset type: SYSRSn
5	RESERVED	R-0	0h	Reserved
4	SDCLKSYNC	R/W	0h	0: SD Clock is not passed through a synchronizer. 1: SD Clock is passed through a synchronizer. Reset type: SYSRSn
3	SDCLKSEL	R/W	0h	SD3 Clock source select. 0: Clock source to SDFM filter is its channel clock. 1: Clock source to SDFM filter is SD1 filter clock. Reset type: SYSRSn
2	RESERVED	R/W	0h	Reserved
1-0	MOD	R/W	0h	Modulator clock modes 0: Mode 0: Modulator clock running at 1x data rate 1: Reserved 2: Reserved 3: Reserved Reset type: SYSRSn

32.12.2.36 SDDFPARM3 Register (Offset = 62h) [Reset = 0000h]

SDDFPARM3 is shown in Figure 32-50 and described in Table 32-48.

Return to the Summary Table.

Data Filter Parameter Register for Ch3

Figure 32-50. SDDFPARM3 Register

15	14	13	12	11	10	9	8
RESERVED			SDSYNCEN	SST		AE	FEN
R-0h			R/W-0h	R/W-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DOSR							
R/W-0h							

Table 32-48. SDDFPARM3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	SDSYNCEN	R/W	0h	PWM synchronization (SDSYNC) of data filter 0: PWM synchronization of data filter is disabled 1: PWM synchronization of data filter is enabled Note: SDSYNCx.SYNCSEL bits define which PWM signal is used to synchronize PWMs Reset type: SYSRSn
11-10	SST	R/W	0h	Data filter structure 00: Data filter runs with a Sincfast structure 01: Data filter runs with a Sinc1 structure 10: Data filter runs with a Sinc2 structure 11: Data filter runs with a Sinc3 structure Reset type: SYSRSn
9	AE	R/W	0h	Data filter Acknowledge Enable 0: Acknowledge flag is disabled for the particular filter 1: Acknowledge flag is enabled for the particular filter Reset type: SYSRSn
8	FEN	R/W	0h	Filter Enable 0: The data filter is disabled and no data is produced 1: The data filter is enabled and data are produced in the data filter Note: When filter is disabled, DOSR counter held in reset, filter data erased. Also resets FIFO pointers and clears the FIFO Reset type: SYSRSn
7-0	DOSR	R/W	0h	Data filter Oversampling ratio The actual oversampling ratio of data filter is DOSR + 1 These bits set the oversampling ratio of the data filter. 0x0FF represents an oversampling ratio of 256. Reset type: SYSRSn

32.12.2.37 SDDPARM3 Register (Offset = 64h) [Reset = 0000h]

SDDPARM3 is shown in Figure 32-51 and described in Table 32-49.

Return to the Summary Table.

Data Parameter Register for Ch3

Figure 32-51. SDDPARM3 Register

15	14	13	12	11	10	9	8
SH					DR	RESERVED	
R/W-0h					R/W-0h	R-0-0h	
7	6	5	4	3	2	1	0
RESERVED							
R-0-0h							

Table 32-49. SDDPARM3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	SH	R/W	0h	Shift Control These bits indicate by how many bits the 16-bit window is shifted up when 16-bit data representation is chosen. Reset type: SYSRSn
10	DR	R/W	0h	Data filter Data representation 0: Data stored in 16b 2's complement 1: Data stored in 32b 2's complement Reset type: SYSRSn
9-0	RESERVED	R-0	0h	Reserved

32.12.2.38 SDFLT3CMPH1 Register (Offset = 66h) [Reset = 7FFFh]

SDFLT3CMPH1 is shown in Figure 32-52 and described in Table 32-50.

Return to the Summary Table.

High-level Threshold Register for Ch3

Figure 32-52. SDFLT3CMPH1 Register

15	14	13	12	11	10	9	8
RESERVED	HLT						
R-0-0h	R/W-7FFFh						
7	6	5	4	3	2	1	0
HLT							
R/W-7FFFh							

Table 32-50. SDFLT3CMPH1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	HLT	R/W	7FFFh	Unsigned high-level threshold for the comparator filter output. Reset type: SYSRSn

32.12.2.39 SDFLT3CMPL1 Register (Offset = 68h) [Reset = 0000h]

SDFLT3CMPL1 is shown in Figure 32-53 and described in Table 32-51.

Return to the Summary Table.

Low-level Threshold Register for Ch3

Figure 32-53. SDFLT3CMPL1 Register

15	14	13	12	11	10	9	8
RESERVED	LLT						
R-0-0h	R/W-0h						
7	6	5	4	3	2	1	0
LLT							
R/W-0h							

Table 32-51. SDFLT3CMPL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	LLT	R/W	0h	Unsigned low-level threshold for the comparator filter output. Reset type: SYSRSn

32.12.2.40 SDCPARM3 Register (Offset = 6Ah) [Reset = 2000h]

SDCPARM3 is shown in Figure 32-54 and described in Table 32-52.

Return to the Summary Table.

Comparator Filter Parameter Register for Ch3

Figure 32-54. SDCPARM3 Register

15	14	13	12	11	10	9	8
CEVT2SEL		CEN	CEVT1SEL		HZEN	MFIE	CS1_CS0
R/W-0h		R/W-1h	R/W-0h		R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CS1_CS0	EN_CEVT2	EN_CEVT1	COSR				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				

Table 32-52. SDCPARM3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	CEVT2SEL	R/W	0h	Comparator Event2 Select 00: COMPL1 01: COMPL1 OR COMPH1 10: COMPL2 11: COMPL2 OR COMPH2 Reset type: SYSRSn
13	CEN	R/W	1h	Comparator Filter enable 0: Disable comparator filter 1: Enable comparator filter Reset type: SYSRSn
12-11	CEVT1SEL	R/W	0h	Comparator Event1 Select 00: COMPH1 01: COMPL1 OR COMPH1 10: COMPH2 11: COMPL2 OR COMPH2 Reset type: SYSRSn
10	HZEN	R/W	0h	High level (Z) Threshold crossing output enable 0: Disable Higher level Threshold (Z) crossing 1: Enable Higher level Threhold (Z) crossing Reset type: SYSRSn
9	MFIE	R/W	0h	Modulator Failure Interrupt Enable 0: Disable modulator failure interrupt and its flag 1: Enable modulator failure interrupt and its flag Reset type: SYSRSn
8-7	CS1_CS0	R/W	0h	Comparator filter structure 00: Comparator filter runs with a sincfast structure 01: Comparator filter runs with a Sinc1 structure 10: Comparator filter runs with a Sinc2 structure 11: Comparator filter runs with a Sinc3 structure Reset type: SYSRSn
6	EN_CEVT2	R/W	0h	CEVT2 interrupt enable 0: Disable CEVT2 interrupt 1: Enable CEVT2 interrupt Reset type: SYSRSn
5	EN_CEVT1	R/W	0h	CEVT1 interrupt enable 0: Disable CEVT1 interrupt 1: Enable CEVT1 interrupt Reset type: SYSRSn

Table 32-52. SDCPARM3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	COSR	R/W	0h	Comparator Oversampling ratio. The actual rate is COSR + 1. These bits set the oversampling ratio of the filter. 0x1F represents an oversampling ratio of 32 Reset type: SYSRSn

32.12.2.41 SDDATA3 Register (Offset = 6Ch) [Reset = 00000000h]

SDDATA3 is shown in Figure 32-55 and described in Table 32-53.

Return to the Summary Table.

Data Filter Data Register (16 or 32bit) for Ch3

Figure 32-55. SDDATA3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA32HI																DATA16															
R-0h																R-0h															

Table 32-53. SDDATA3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DATA32HI	R	0h	Hi-order 16b in 32b mode, 16-bit Data in 16b mode Reset type: SYSRSn
15-0	DATA16	R	0h	Lo-order 16b in 32b mode Reset type: SYSRSn

32.12.2.42 SDDATFIFO3 Register (Offset = 70h) [Reset = 00000000h]

SDDATFIFO3 is shown in Figure 32-56 and described in Table 32-54.

Return to the Summary Table.

Filter Data FIFO Output(32b) for Ch3

Figure 32-56. SDDATFIFO3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA32HI																DATA16															
R-0h																R-0h															

Table 32-54. SDDATFIFO3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DATA32HI	R	0h	Hi-order 16b in 32b mode, 16-bit Data in 16b mode Reset type: SYSRSn
15-0	DATA16	R	0h	Lo-order 16b in 32b mode Reset type: SYSRSn

32.12.2.43 SDCDATA3 Register (Offset = 74h) [Reset = 0000h]

SDCDATA3 is shown in Figure 32-57 and described in Table 32-55.

Return to the Summary Table.

Comparator Filter Data Register (16b) for Ch3

Figure 32-57. SDCDATA3 Register

15	14	13	12	11	10	9	8
DATA16							
R-0h							
7	6	5	4	3	2	1	0
DATA16							
R-0h							

Table 32-55. SDCDATA3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	DATA16	R	0h	Comparator Data output - 16b only Reset type: SYSRSn

32.12.2.44 SDFLT3CMPH2 Register (Offset = 76h) [Reset = 7FFFh]

SDFLT3CMPH2 is shown in Figure 32-58 and described in Table 32-56.

Return to the Summary Table.

Second high level threhold for CH3

Figure 32-58. SDFLT3CMPH2 Register

15	14	13	12	11	10	9	8
RESERVED	HLT2						
R-0-0h	R/W-7FFFh						
7	6	5	4	3	2	1	0
HLT2							
R/W-7FFFh							

Table 32-56. SDFLT3CMPH2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	HLT2	R/W	7FFFh	Second Unsigned high-level threshold for the comparator filter output. Reset type: SYSRSn

32.12.2.45 SDFLT3CMPHZ Register (Offset = 78h) [Reset = 0000h]

SDFLT3CMPHZ is shown in Figure 32-59 and described in Table 32-57.

Return to the Summary Table.

High-level (Z) Threshold Register for Ch3

Figure 32-59. SDFLT3CMPHZ Register

15	14	13	12	11	10	9	8
RESERVED	HLTZ						
R-0-0h	R/W-0h						
7	6	5	4	3	2	1	0
HLTZ							
R/W-0h							

Table 32-57. SDFLT3CMPHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	HLTZ	R/W	0h	Unsigned High-level threshold (Z) for the comparator filter output Primarily intended for detecting 'zero'-crossing events. Unlike the primary comparator SDCMPHx, it does not have the ability to generate an interrupt. Reset type: SYSRSn

32.12.2.46 SDFIFOCTL3 Register (Offset = 7Ah) [Reset = 0000h]

SDFIFOCTL3 is shown in Figure 32-60 and described in Table 32-58.

Return to the Summary Table.

FIFO Control Register for Ch3

Figure 32-60. SDFIFOCTL3 Register

15	14	13	12	11	10	9	8
OVFIEN	DRINTSEL	FFEN	FFIEN	RESERVED	SDFFST		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0-0h	R-0h		
7	6	5	4	3	2	1	0
SDFFST		RESERVED	SDFFIL				
R-0h		R-0-0h	R/W-0h				

Table 32-58. SDFIFOCTL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	OVFIEN	R/W	0h	SDFIFO Overflow interrupt enable 0: SDFIFO Overflow condition will not generate an interrupt 1: SDFIFO overflow condition generates an interrupt on SDy_ERR Reset type: SYSRSn
14	DRINTSEL	R/W	0h	Data-Ready Interrupt (DRINT) source select 0 = AF3 (Select non-FIFO data-ready interrupt) 1 = SDFFINT3 (Select FIFO data-ready interrupt) Reset type: SYSRSn
13	FFEN	R/W	0h	SDFIFO Enable 0: Disable FIFO operation 1: Enable FIFO operation Note: When FIFO is disabled, FIFO contents are cleared Reset type: SYSRSn
12	FFIEN	R/W	0h	SDFIFO data ready Interrupt Enable Reset type: SYSRSn
11	RESERVED	R-0	0h	Reserved
10-6	SDFFST	R	0h	SDFIFO Status 00000 FIFO empty 00001 FIFO has 1 word . . . . 10000 FIFO has 16 words Reset type: SYSRSn
5	RESERVED	R-0	0h	Reserved
4-0	SDFFIL	R/W	0h	SDFIFO interrupt level bits The FIFO will generate an interrupt when the FIFO status (SDFFST) >= FIFO level (SDFFIL) Reset type: SYSRSn

32.12.2.47 SDSYNC3 Register (Offset = 7Ch) [Reset = 043Fh]

SDSYNC3 is shown in Figure 32-61 and described in Table 32-59.

Return to the Summary Table.

SD Filter Sync control for Ch3

Figure 32-61. SDSYNC3 Register

15	14	13	12	11	10	9	8
RESERVED					WTSCLREN	FFSYNCCLREN	WTSYNCLR
R-0-0h					R/W-1h	R/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
WTSYNFLG	WTSYNCEN	SYNCSEL					
R-0h	R/W-0h	R/W-3Fh					

Table 32-59. SDSYNC3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R-0	0h	Reserved
10	WTSCLREN	R/W	1h	WTSYNFLG Clear-on-FIFOINT Enable 0: WTSYNFLG can only be cleared manually (using WTSYNCLR bit) 1: WTSYNFLG is cleared automatically on SDFFINT Reset type: SYSRSn
9	FFSYNCCLREN	R/W	0h	FIFO Clear-on-SDSYNC Enable 0: SDFIFO is not automaticaly cleared upon receiving SDSYNC 1: SDFIFO is automaticaly cleared upon receiving SDSYNC Reset type: SYSRSn
8	WTSYNCLR	R-0/W	0h	Wait-for-Sync Flag Clear (always reads 0) 0: Write of 0 has no affect 1: Write of 1 clears WTSYNFLG Reset type: SYSRSn
7	WTSYNFLG	R	0h	Wait-for-Sync Flag 0: SDSYNC event has not occurred 1: SDSYNC event occurred. Reset type: SYSRSn
6	WTSYNCEN	R/W	0h	Wait-for-Sync Enable 0: Incoming Data written to SDFIFO on every Data-Ready (DR) Event 1: Incoming Data written to SDFIFO on DR event only after SDSYNC event occurs Reset type: SYSRSn
5-0	SYNCSEL	R/W	3Fh	Defines source for the SDSYNC Input on this channel Refer SDSYNCx.SYNCSEL table Reset type: SYSRSn

32.12.2.48 SDFLT3CMPL2 Register (Offset = 7Eh) [Reset = 0000h]

SDFLT3CMPL2 is shown in Figure 32-62 and described in Table 32-60.

Return to the Summary Table.

Second low level threhold for CH3

Figure 32-62. SDFLT3CMPL2 Register

15	14	13	12	11	10	9	8
RESERVED	LLT2						
R-0-0h	R/W-0h						
7	6	5	4	3	2	1	0
LLT2							
R/W-0h							

Table 32-60. SDFLT3CMPL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	LLT2	R/W	0h	Second Unsigned low-level threshold for the comparator filter output. Reset type: SYSRSn

32.12.2.49 SDCTLPARM4 Register (Offset = 80h) [Reset = 0000h]

SDCTLPARM4 is shown in Figure 32-63 and described in Table 32-61.

Return to the Summary Table.

Control Parameter Register for Ch4

Figure 32-63. SDCTLPARM4 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	SDDATASYNC	RESERVED	SDCLKSYNC	SDCLKSEL	RESERVED	MOD	
R-0-0h	R/W-0h	R-0-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

Table 32-61. SDCTLPARM4 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	RESERVED	R-0	0h	Reserved
6	SDDATASYNC	R/W	0h	0: SD Data is not passed through a synchronizer. 1: SD Data is passed through a synchronizer. Reset type: SYSRSn
5	RESERVED	R-0	0h	Reserved
4	SDCLKSYNC	R/W	0h	0: SD Clock is not passed through a synchronizer. 1: SD Clock is passed through a synchronizer. Reset type: SYSRSn
3	SDCLKSEL	R/W	0h	SD4 Clock source select. 0: Clock source to SDFM filter is its channel clock. 1: Clock source to SDFM filter is SD1 filter clock. Reset type: SYSRSn
2	RESERVED	R/W	0h	Reserved
1-0	MOD	R/W	0h	Modulator clock modes 0: Mode 0: Modulator clock running at 1x data rate 1: Reserved 2: Reserved 3: Reserved Reset type: SYSRSn

32.12.2.50 SDDFPARM4 Register (Offset = 82h) [Reset = 0000h]

SDDFPARM4 is shown in Figure 32-64 and described in Table 32-62.

Return to the Summary Table.

Data Filter Parameter Register for Ch4

Figure 32-64. SDDFPARM4 Register

15	14	13	12	11	10	9	8
RESERVED			SDSYNCEN	SST		AE	FEN
R-0h			R/W-0h	R/W-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DOSR							
R/W-0h							

Table 32-62. SDDFPARM4 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	SDSYNCEN	R/W	0h	PWM synchronization (SDSYNC) of data filter 0: PWM synchronization of data filter is disabled 1: PWM synchronization of data filter is enabled Note: SDSYNCx.SYNCSEL bits define which PWM signal is used to synchronize PWMs Reset type: SYSRSn
11-10	SST	R/W	0h	Data filter structure 00: Data filter runs with a Sincfast structure 01: Data filter runs with a Sinc1 structure 10: Data filter runs with a Sinc2 structure 11: Data filter runs with a Sinc3 structure Reset type: SYSRSn
9	AE	R/W	0h	Data filter Acknowledge Enable 0: Acknowledge flag is disabled for the particular filter 1: Acknowledge flag is enabled for the particular filter Reset type: SYSRSn
8	FEN	R/W	0h	Filter Enable 0: The data filter is disabled and no data is produced 1: The data filter is enabled and data are produced in the data filter Note: When filter is disabled, DOSR counter held in reset, filter data erased. Also resets FIFO pointers and clears the FIFO Reset type: SYSRSn
7-0	DOSR	R/W	0h	Data filter Oversampling ratio The actual oversampling ratio of data filter is DOSR + 1 These bits set the oversampling ratio of the data filter. 0x0FF represents an oversampling ratio of 256. Reset type: SYSRSn

32.12.2.51 SDDPARM4 Register (Offset = 84h) [Reset = 0000h]

SDDPARM4 is shown in Figure 32-65 and described in Table 32-63.

Return to the Summary Table.

Data Parameter Register for Ch4

Figure 32-65. SDDPARM4 Register

15	14	13	12	11	10	9	8
SH					DR	RESERVED	
R/W-0h					R/W-0h	R-0-0h	
7	6	5	4	3	2	1	0
RESERVED							
R-0-0h							

Table 32-63. SDDPARM4 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	SH	R/W	0h	Shift Control These bits indicate by how many bits the 16-bit window is shifted up when 16-bit data representation is chosen. Reset type: SYSRSn
10	DR	R/W	0h	Data filter Data representation 0: Data stored in 16b 2's complement 1: Data stored in 32b 2's complement Reset type: SYSRSn
9-0	RESERVED	R-0	0h	Reserved

32.12.2.52 SDFLT4CMPH1 Register (Offset = 86h) [Reset = 7FFFh]

SDFLT4CMPH1 is shown in Figure 32-66 and described in Table 32-64.

Return to the Summary Table.

High-level Threshold Register for Ch4

Figure 32-66. SDFLT4CMPH1 Register

15	14	13	12	11	10	9	8
RESERVED	HLT						
R-0-0h	R/W-7FFFh						
7	6	5	4	3	2	1	0
HLT							
R/W-7FFFh							

Table 32-64. SDFLT4CMPH1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	HLT	R/W	7FFFh	Unsigned high-level threshold for the comparator filter output. Reset type: SYSRSn

32.12.2.53 SDFLT4CMPL1 Register (Offset = 88h) [Reset = 0000h]

SDFLT4CMPL1 is shown in Figure 32-67 and described in Table 32-65.

Return to the Summary Table.

Low-level Threshold Register for Ch4

Figure 32-67. SDFLT4CMPL1 Register

15	14	13	12	11	10	9	8
RESERVED	LLT						
R-0-0h	R/W-0h						
7	6	5	4	3	2	1	0
LLT							
R/W-0h							

Table 32-65. SDFLT4CMPL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	LLT	R/W	0h	Unsigned low-level threshold for the comparator filter output. Reset type: SYSRSn

32.12.2.54 SDCPARM4 Register (Offset = 8Ah) [Reset = 2000h]

SDCPARM4 is shown in Figure 32-68 and described in Table 32-66.

Return to the Summary Table.

Comparator Filter Parameter Register for Ch4

Figure 32-68. SDCPARM4 Register

15	14	13	12	11	10	9	8
CEVT2SEL		CEN	CEVT1SEL		HZEN	MFIE	CS1_CS0
R/W-0h		R/W-1h	R/W-0h		R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CS1_CS0	EN_CEVT2	EN_CEVT1	COSR				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				

Table 32-66. SDCPARM4 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	CEVT2SEL	R/W	0h	Comparator Event2 Select 00: COMPL1 01: COMPL1 OR COMPH1 10: COMPL2 11: COMPL2 OR COMPH2 Reset type: SYSRSn
13	CEN	R/W	1h	Comparator Filter enable 0: Disable comparator filter 1: Enable comparator filter Reset type: SYSRSn
12-11	CEVT1SEL	R/W	0h	Comparator Event1 Select 00: COMPH1 01: COMPL1 OR COMPH1 10: COMPH2 11: COMPL2 OR COMPH2 Reset type: SYSRSn
10	HZEN	R/W	0h	High level (Z) Threshold crossing output enable 0: Disable Higher level Threshold (Z) crossing 1: Enable Higher level Threhold (Z) crossing Reset type: SYSRSn
9	MFIE	R/W	0h	Modulator Failure Interrupt Enable 0: Disable modulator failure interrupt and its flag 1: Enable modulator failure interrupt and its flag Reset type: SYSRSn
8-7	CS1_CS0	R/W	0h	Comparator filter structure 00: Comparator filter runs with a sincfast structure 01: Comparator filter runs with a Sinc1 structure 10: Comparator filter runs with a Sinc2 structure 11: Comparator filter runs with a Sinc3 structure Reset type: SYSRSn
6	EN_CEVT2	R/W	0h	CEVT2 interrupt enable 0: Disable CEVT2 interrupt 1: Enable CEVT2 interrupt Reset type: SYSRSn
5	EN_CEVT1	R/W	0h	CEVT1 interrupt enable 0: Disable CEVT1 interrupt 1: Enable CEVT1 interrupt Reset type: SYSRSn

Table 32-66. SDCPARM4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	COSR	R/W	0h	Comparator Oversampling ratio. The actual rate is COSR + 1. These bits set the oversampling ratio of the filter. 0x1F represents an oversampling ratio of 32 Reset type: SYSRSn

32.12.2.55 SDDATA4 Register (Offset = 8Ch) [Reset = 00000000h]

SDDATA4 is shown in Figure 32-69 and described in Table 32-67.

Return to the Summary Table.

Data Filter Data Register (16 or 32bit) for Ch4

Figure 32-69. SDDATA4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA32HI																DATA16															
R-0h																R-0h															

Table 32-67. SDDATA4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DATA32HI	R	0h	Hi-order 16b in 32b mode, 16-bit Data in 16b mode Reset type: SYSRSn
15-0	DATA16	R	0h	Lo-order 16b in 32b mode Reset type: SYSRSn

32.12.2.56 SDDATFIFO4 Register (Offset = 90h) [Reset = 00000000h]

SDDATFIFO4 is shown in Figure 32-70 and described in Table 32-68.

Return to the Summary Table.

Filter Data FIFO Output(32b) for Ch4

Figure 32-70. SDDATFIFO4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA32HI																DATA16															
R-0h																R-0h															

Table 32-68. SDDATFIFO4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DATA32HI	R	0h	Hi-order 16b in 32b mode, 16-bit Data in 16b mode Reset type: SYSRSn
15-0	DATA16	R	0h	Lo-order 16b in 32b mode Reset type: SYSRSn

32.12.2.57 SDCDATA4 Register (Offset = 94h) [Reset = 0000h]

SDCDATA4 is shown in Figure 32-71 and described in Table 32-69.

Return to the Summary Table.

Comparator Filter Data Register (16b) for Ch4

Figure 32-71. SDCDATA4 Register

15	14	13	12	11	10	9	8
DATA16							
R-0h							
7	6	5	4	3	2	1	0
DATA16							
R-0h							

Table 32-69. SDCDATA4 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	DATA16	R	0h	Comparator Data output - 16b only Reset type: SYSRSn

32.12.2.58 SDFLT4CMPH2 Register (Offset = 96h) [Reset = 7FFFh]

SDFLT4CMPH2 is shown in Figure 32-72 and described in Table 32-70.

Return to the Summary Table.

Second high level threhold for CH4

Figure 32-72. SDFLT4CMPH2 Register

15	14	13	12	11	10	9	8
RESERVED	HLT2						
R-0-0h	R/W-7FFFh						
7	6	5	4	3	2	1	0
HLT2							
R/W-7FFFh							

Table 32-70. SDFLT4CMPH2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	HLT2	R/W	7FFFh	Second Unsigned high-level threshold for the comparator filter output. Reset type: SYSRSn

32.12.2.59 SDFLT4CMPHZ Register (Offset = 98h) [Reset = 0000h]

SDFLT4CMPHZ is shown in Figure 32-73 and described in Table 32-71.

Return to the Summary Table.

High-level (Z) Threshold Register for Ch4

Figure 32-73. SDFLT4CMPHZ Register

15	14	13	12	11	10	9	8
RESERVED	HLTZ						
R-0-0h	R/W-0h						
7	6	5	4	3	2	1	0
HLTZ							
R/W-0h							

Table 32-71. SDFLT4CMPHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	HLTZ	R/W	0h	Unsigned High-level threshold (Z) for the comparator filter output Primarily intended for detecting 'zero'-crossing events. Unlike the primary comparator SDCMPHx, it does not have the ability to generate an interrupt. Reset type: SYSRSn

32.12.2.60 SDFIFOCTL4 Register (Offset = 9Ah) [Reset = 0000h]

SDFIFOCTL4 is shown in Figure 32-74 and described in Table 32-72.

Return to the Summary Table.

FIFO Control Register for Ch4

Figure 32-74. SDFIFOCTL4 Register

15	14	13	12	11	10	9	8
OVFIEN	DRINTSEL	FFEN	FFIEN	RESERVED	SDFFST		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0-0h	R-0h		
7	6	5	4	3	2	1	0
SDFFST		RESERVED	SDFFIL				
R-0h		R-0-0h	R/W-0h				

Table 32-72. SDFIFOCTL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	OVFIEN	R/W	0h	SDFIFO Overflow interrupt enable 0: SDFIFO Overflow condition will not generate an interrupt 1: SDFIFO overflow condition generates an interrupt on SDy_ERR Reset type: SYSRSn
14	DRINTSEL	R/W	0h	Data-Ready Interrupt (DRINT) source select 0 = AF4 (Select non-FIFO data-ready interrupt) 1 = SDFFINT4 (Select FIFO data-ready interrupt) Reset type: SYSRSn
13	FFEN	R/W	0h	SDFIFO Enable 0: Disable FIFO operation 1: Enable FIFO operation Note: When FIFO is disabled, FIFO contents are cleared Reset type: SYSRSn
12	FFIEN	R/W	0h	SDFIFO data ready Interrupt Enable Reset type: SYSRSn
11	RESERVED	R-0	0h	Reserved
10-6	SDFFST	R	0h	SDFIFO Status 00000 FIFO empty 00001 FIFO has 1 word . . . . 10000 FIFO has 16 words Reset type: SYSRSn
5	RESERVED	R-0	0h	Reserved
4-0	SDFFIL	R/W	0h	SDFIFO interrupt level bits The FIFO will generate an interrupt when the FIFO status (SDFFST) >= FIFO level (SDFFIL) Reset type: SYSRSn

32.12.2.61 SDSYNC4 Register (Offset = 9Ch) [Reset = 043Fh]

SDSYNC4 is shown in Figure 32-75 and described in Table 32-73.

Return to the Summary Table.

SD Filter Sync control for Ch4

Figure 32-75. SDSYNC4 Register

15	14	13	12	11	10	9	8
RESERVED					WTSCLREN	FFSYNCCLREN	WTSYNCLR
R-0-0h					R/W-1h	R/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
WTSYNFLG	WTSYNCEN	SYNCSEL					
R-0h	R/W-0h	R/W-3Fh					

Table 32-73. SDSYNC4 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R-0	0h	Reserved
10	WTSCLREN	R/W	1h	WTSYNFLG Clear-on-FIFOINT Enable 0: WTSYNFLG can only be cleared manually (using WTSYNCLR bit) 1: WTSYNFLG is cleared automatically on SDFFINT Reset type: SYSRSn
9	FFSYNCCLREN	R/W	0h	FIFO Clear-on-SDSYNC Enable 0: SDFIFO is not automaticaly cleared upon receiving SDSYNC 1: SDFIFO is automaticaly cleared upon receiving SDSYNC Reset type: SYSRSn
8	WTSYNCLR	R-0/W	0h	Wait-for-Sync Flag Clear (always reads 0) 0: Write of 0 has no affect 1: Write of 1 clears WTSYNFLG Reset type: SYSRSn
7	WTSYNFLG	R	0h	Wait-for-Sync Flag 0: SDSYNC event has not occurred 1: SDSYNC event occurred. Reset type: SYSRSn
6	WTSYNCEN	R/W	0h	Wait-for-Sync Enable 0: Incoming Data written to SDFIFO on every Data-Ready (DR) Event 1: Incoming Data written to SDFIFO on DR event only after SDSYNC event occurs Reset type: SYSRSn
5-0	SYNCSEL	R/W	3Fh	Defines source for the SDSYNC Input on this channel Refer SDSYNCx.SYNCSEL table Reset type: SYSRSn

32.12.2.62 SDFLT4CMPL2 Register (Offset = 9Eh) [Reset = 0000h]

SDFLT4CMPL2 is shown in Figure 32-76 and described in Table 32-74.

Return to the Summary Table.

Second low level threhold for CH4

Figure 32-76. SDFLT4CMPL2 Register

15	14	13	12	11	10	9	8
RESERVED	LLT2						
R-0-0h	R/W-0h						
7	6	5	4	3	2	1	0
LLT2							
R/W-0h							

Table 32-74. SDFLT4CMPL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14-0	LLT2	R/W	0h	Second Unsigned low-level threshold for the comparator filter output. Reset type: SYSRSn

32.12.2.63 SDCOMP1CTL Register (Offset = C0h) [Reset = 0000h]

SDCOMP1CTL is shown in Figure 32-77 and described in Table 32-75.

Return to the Summary Table.

SD Comparator event filter1 Control Register

Figure 32-77. SDCOMP1CTL Register

15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED		CEVT2DIGFILTSEL		RESERVED	RESERVED
R-0h	R-0h	R-0h		R/W-0h		R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED		CEVT1DIGFILTSEL		RESERVED	RESERVED
R-0h	R-0h	R-0h		R/W-0h		R-0h	R-0h

Table 32-75. SDCOMP1CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13-12	RESERVED	R	0h	Reserved
11-10	CEVT2DIGFILTSEL	R/W	0h	High comparator COMPH source select. 0 CEVT2 output drives COMPLOUT 1 Reserved 2 Output of digital filter drives COMPLOUT 3 Reserved Reset type: SYSRSn
9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5-4	RESERVED	R	0h	Reserved
3-2	CEVT1DIGFILTSEL	R/W	0h	High comparator COMPH source select. 0 CEVT1 output drives COMPHOUT 1 Reserved 2 Output of digital filter drives COMPHOUT 3 Reserved Reset type: SYSRSn
1	RESERVED	R	0h	Reserved
0	RESERVED	R	0h	Reserved

32.12.2.64 SDCOMP1EVT2FLTCTL Register (Offset = C2h) [Reset = 0000h]

SDCOMP1EVT2FLTCTL is shown in Figure 32-78 and described in Table 32-76.

Return to the Summary Table.

COMPL/CEVT2 Digital filter1 Control Register

Figure 32-78. SDCOMP1EVT2FLTCTL Register

15	14	13	12	11	10	9	8
FILINIT	RESERVED	THRESH					SAMPWIN
R-0/W1S-0h	R-0h	R/W-0h					R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN				RESERVED			
R/W-0h				R-0h			

Table 32-76. SDCOMP1EVT2FLTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	Low filter initialization. 0 No effect 1 Initialize all samples to the filter input value Reset type: SYSRSn
14	RESERVED	R	0h	Reserved
13-9	THRESH	R/W	0h	Low filter majority voting threshold. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Reset type: SYSRSn
8-4	SAMPWIN	R/W	0h	Low filter sample window size. Number of samples to monitor is SAMPWIN+1. Reset type: SYSRSn
3-0	RESERVED	R	0h	Reserved

32.12.2.65 SDCOMP1EVT2FLTCLKCTL Register (Offset = C4h) [Reset = 0000h]

SDCOMP1EVT2FLTCLKCTL is shown in Figure 32-79 and described in Table 32-77.

Return to the Summary Table.

COMPL/CEVT2 Digital filter1 Clock Control Register

Figure 32-79. SDCOMP1EVT2FLTCLKCTL Register

15	14	13	12	11	10	9	8
RESERVED						CLKPRESCALE	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
CLKPRESCALE							
R/W-0h							

Table 32-77. SDCOMP1EVT2FLTCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	CLKPRESCALE	R/W	0h	Low filter sample clock prescale. Number of system clocks between samples. Reset type: SYSRSn

32.12.2.66 SDCOMP1EVT1FLTCTL Register (Offset = C6h) [Reset = 0000h]

SDCOMP1EVT1FLTCTL is shown in Figure 32-80 and described in Table 32-78.

Return to the Summary Table.

COMPH/CEVT1 Digital filter1 Control Register

Figure 32-80. SDCOMP1EVT1FLTCTL Register

15	14	13	12	11	10	9	8
FILINIT	RESERVED	THRESH					SAMPWIN
R-0/W1S-0h	R-0h	R/W-0h					R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN				RESERVED			
R/W-0h				R-0h			

Table 32-78. SDCOMP1EVT1FLTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	High filter initialization. 0 No effect 1 Initialize all samples to the filter input value Reset type: SYSRSn
14	RESERVED	R	0h	Reserved
13-9	THRESH	R/W	0h	High filter majority voting threshold. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Reset type: SYSRSn
8-4	SAMPWIN	R/W	0h	High filter sample window size. Number of samples to monitor is SAMPWIN+1. Reset type: SYSRSn
3-0	RESERVED	R	0h	Reserved

32.12.2.67 SDCOMP1EVT1FLTCLKCTL Register (Offset = C8h) [Reset = 0000h]

SDCOMP1EVT1FLTCLKCTL is shown in Figure 32-81 and described in Table 32-79.

Return to the Summary Table.

COMPH/CEVT1 Digital filter1 Clock Control Register

Figure 32-81. SDCOMP1EVT1FLTCLKCTL Register

15	14	13	12	11	10	9	8
RESERVED						CLKPRESCALE	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
CLKPRESCALE							
R/W-0h							

Table 32-79. SDCOMP1EVT1FLTCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	CLKPRESCALE	R/W	0h	High filter sample clock prescale. Number of system clocks between samples. Reset type: SYSRSn

32.12.2.68 SDCOMP1LOCK Register (Offset = CEh) [Reset = 0000h]

SDCOMP1LOCK is shown in Figure 32-82 and described in Table 32-80.

Return to the Summary Table.

SD compartor event filter1 Lock Register

Figure 32-82. SDCOMP1LOCK Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			RESERVED	COMP	RESERVED	RESERVED	SDCOMP1CTL
R-0h			R-0h	R/WSonce-0h	R-0h	R-0h	R/WSonce-0h

Table 32-80. SDCOMP1LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	COMP	R/WSonce	0h	Lock write-access to the SDCOMP1EVT1/2FLTTCTL and COMP1FILCLKCTL registers. 0 SDCOMP1EVT1/2FLTCTL and SDCOMP1EVT1/2FLTCLKCTL registers are not locked. Write 0 to this bit has no effect. 1 SDCOMP1EVT1/2FLTCTL and SDCOMP1EVT1/2FLTCLKCTL registers are locked. Only a system reset can clear this bit. Reset type: SYSRSn
2	RESERVED	R	0h	Reserved
1	RESERVED	R	0h	Reserved
0	SDCOMP1CTL	R/WSonce	0h	Lock write-access to the SDCOMP1CTL register. 0 SDCOMP1CTL register is not locked. Write 0 to this bit has no effect. 1 SDCOMP1CTL register is locked. Only a system reset can clear this bit. Reset type: SYSRSn

32.12.2.69 SDCOMP2CTL Register (Offset = D0h) [Reset = 0000h]

SDCOMP2CTL is shown in Figure 32-83 and described in Table 32-81.

Return to the Summary Table.

SD Comparator event filter2 Control Register

Figure 32-83. SDCOMP2CTL Register

15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED		CEVT2DIGFILTSEL		RESERVED	RESERVED
R-0h	R-0h	R-0h		R/W-0h		R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED		CEVT1DIGFILTSEL		RESERVED	RESERVED
R-0h	R-0h	R-0h		R/W-0h		R-0h	R-0h

Table 32-81. SDCOMP2CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13-12	RESERVED	R	0h	Reserved
11-10	CEVT2DIGFILTSEL	R/W	0h	High comparator COMPH source select. 0 CEVT2 output drives COMPLOUT 1 Reserved 2 Output of digital filter drives COMPLOUT 3 Reserved Reset type: SYSRSn
9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5-4	RESERVED	R	0h	Reserved
3-2	CEVT1DIGFILTSEL	R/W	0h	High comparator COMPH source select. 0 CEVT1 output drives COMPHOUT 1 Reserved 2 Output of digital filter drives COMPHOUT 3 Reserved Reset type: SYSRSn
1	RESERVED	R	0h	Reserved
0	RESERVED	R	0h	Reserved

32.12.2.70 SDCOMP2EVT2FLTCTL Register (Offset = D2h) [Reset = 0000h]

SDCOMP2EVT2FLTCTL is shown in Figure 32-84 and described in Table 32-82.

Return to the Summary Table.

COMPL/CEVT2 Digital filter2 Control Register

Figure 32-84. SDCOMP2EVT2FLTCTL Register

15	14	13	12	11	10	9	8
FILINIT	RESERVED	THRESH					SAMPWIN
R-0/W1S-0h	R-0h	R/W-0h					R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN				RESERVED			
R/W-0h				R-0h			

Table 32-82. SDCOMP2EVT2FLTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	Low filter initialization. 0 No effect 1 Initialize all samples to the filter input value Reset type: SYSRSn
14	RESERVED	R	0h	Reserved
13-9	THRESH	R/W	0h	Low filter majority voting threshold. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Reset type: SYSRSn
8-4	SAMPWIN	R/W	0h	Low filter sample window size. Number of samples to monitor is SAMPWIN+1. Reset type: SYSRSn
3-0	RESERVED	R	0h	Reserved

32.12.2.71 SDCOMP2EVT2FLTCLKCTL Register (Offset = D4h) [Reset = 0000h]

SDCOMP2EVT2FLTCLKCTL is shown in Figure 32-85 and described in Table 32-83.

Return to the Summary Table.

COMPL/CEVT2 Digital filter2 Clock Control Register

Figure 32-85. SDCOMP2EVT2FLTCLKCTL Register

15	14	13	12	11	10	9	8
RESERVED						CLKPRESCALE	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
CLKPRESCALE							
R/W-0h							

Table 32-83. SDCOMP2EVT2FLTCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	CLKPRESCALE	R/W	0h	Low filter sample clock prescale. Number of system clocks between samples. Reset type: SYSRSn

32.12.2.72 SDCOMP2EVT1FLTCTL Register (Offset = D6h) [Reset = 0000h]

SDCOMP2EVT1FLTCTL is shown in Figure 32-86 and described in Table 32-84.

Return to the Summary Table.

COMPH/CEVT1 Digital filter2 Control Register

Figure 32-86. SDCOMP2EVT1FLTCTL Register

15	14	13	12	11	10	9	8
FILINIT	RESERVED	THRESH					SAMPWIN
R-0/W1S-0h	R-0h	R/W-0h					R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN				RESERVED			
R/W-0h				R-0h			

Table 32-84. SDCOMP2EVT1FLTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	High filter initialization. 0 No effect 1 Initialize all samples to the filter input value Reset type: SYSRSn
14	RESERVED	R	0h	Reserved
13-9	THRESH	R/W	0h	High filter majority voting threshold. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Reset type: SYSRSn
8-4	SAMPWIN	R/W	0h	High filter sample window size. Number of samples to monitor is SAMPWIN+1. Reset type: SYSRSn
3-0	RESERVED	R	0h	Reserved

32.12.2.73 SDCOMP2EVT1FLTCLKCTL Register (Offset = D8h) [Reset = 0000h]

SDCOMP2EVT1FLTCLKCTL is shown in Figure 32-87 and described in Table 32-85.

Return to the Summary Table.

COMPH/CEVT1 Digital filter2 Clock Control Register

Figure 32-87. SDCOMP2EVT1FLTCLKCTL Register

15	14	13	12	11	10	9	8
RESERVED						CLKPRESCALE	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
CLKPRESCALE							
R/W-0h							

Table 32-85. SDCOMP2EVT1FLTCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	CLKPRESCALE	R/W	0h	High filter sample clock prescale. Number of system clocks between samples. Reset type: SYSRSn

32.12.2.74 SDCOMP2LOCK Register (Offset = DEh) [Reset = 0000h]

SDCOMP2LOCK is shown in Figure 32-88 and described in Table 32-86.

Return to the Summary Table.

SD compartor event filter2 Lock Register

Figure 32-88. SDCOMP2LOCK Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			RESERVED	COMP	RESERVED	RESERVED	SDCOMP2CTL
R-0h			R-0h	R/WSonce-0h	R-0h	R-0h	R/WSonce-0h

Table 32-86. SDCOMP2LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	COMP	R/WSonce	0h	Lock write-access to the SDCOMP2EVT1/2FLTTCTL and COMP2FILCLKCTL registers. 0 SDCOMP2EVT1/2FLTCTL and SDCOMP2EVT1/2FLTCLKCTL registers are not locked. Write 0 to this bit has no effect. 1 SDCOMP2EVT1/2FLTCTL and SDCOMP2EVT1/2FLTCLKCTL registers are locked. Only a system reset can clear this bit. Reset type: SYSRSn
2	RESERVED	R	0h	Reserved
1	RESERVED	R	0h	Reserved
0	SDCOMP2CTL	R/WSonce	0h	Lock write-access to the SDCOMP2CTL register. 0 SDCOMP2CTL register is not locked. Write 0 to this bit has no effect. 1 SDCOMP2CTL register is locked. Only a system reset can clear this bit. Reset type: SYSRSn

32.12.2.75 SDCOMP3CTL Register (Offset = E0h) [Reset = 0000h]

SDCOMP3CTL is shown in Figure 32-89 and described in Table 32-87.

Return to the Summary Table.

SD Comparator event filter3 Control Register

Figure 32-89. SDCOMP3CTL Register

15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED		CEVT2DIGFILTSEL		RESERVED	RESERVED
R-0h	R-0h	R-0h		R/W-0h		R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED		CEVT1DIGFILTSEL		RESERVED	RESERVED
R-0h	R-0h	R-0h		R/W-0h		R-0h	R-0h

Table 32-87. SDCOMP3CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13-12	RESERVED	R	0h	Reserved
11-10	CEVT2DIGFILTSEL	R/W	0h	High comparator COMPH source select. 0 CEVT2 output drives COMPLOUT 1 Reserved 2 Output of digital filter drives COMPLOUT 3 Reserved Reset type: SYSRSn
9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5-4	RESERVED	R	0h	Reserved
3-2	CEVT1DIGFILTSEL	R/W	0h	High comparator COMPH source select. 0 CEVT1 output drives COMPHOUT 1 Reserved 2 Output of digital filter drives COMPHOUT 3 Reserved Reset type: SYSRSn
1	RESERVED	R	0h	Reserved
0	RESERVED	R	0h	Reserved

32.12.2.76 SDCOMP3EVT2FLTCTL Register (Offset = E2h) [Reset = 0000h]

SDCOMP3EVT2FLTCTL is shown in Figure 32-90 and described in Table 32-88.

Return to the Summary Table.

COMPL/CEVT2 Digital filter3 Control Register

Figure 32-90. SDCOMP3EVT2FLTCTL Register

15	14	13	12	11	10	9	8
FILINIT	RESERVED	THRESH					SAMPWIN
R-0/W1S-0h	R-0h	R/W-0h					R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN				RESERVED			
R/W-0h				R-0h			

Table 32-88. SDCOMP3EVT2FLTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	Low filter initialization. 0 No effect 1 Initialize all samples to the filter input value Reset type: SYSRSn
14	RESERVED	R	0h	Reserved
13-9	THRESH	R/W	0h	Low filter majority voting threshold. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Reset type: SYSRSn
8-4	SAMPWIN	R/W	0h	Low filter sample window size. Number of samples to monitor is SAMPWIN+1. Reset type: SYSRSn
3-0	RESERVED	R	0h	Reserved

32.12.2.77 SDCOMP3EVT2FLTCLKCTL Register (Offset = E4h) [Reset = 0000h]

SDCOMP3EVT2FLTCLKCTL is shown in Figure 32-91 and described in Table 32-89.

Return to the Summary Table.

COMPL/CEVT2 Digital filter3 Clock Control Register

Figure 32-91. SDCOMP3EVT2FLTCLKCTL Register

15	14	13	12	11	10	9	8
RESERVED						CLKPRESCALE	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
CLKPRESCALE							
R/W-0h							

Table 32-89. SDCOMP3EVT2FLTCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	CLKPRESCALE	R/W	0h	Low filter sample clock prescale. Number of system clocks between samples. Reset type: SYSRSn

32.12.2.78 SDCOMP3EVT1FLTCTL Register (Offset = E6h) [Reset = 0000h]

SDCOMP3EVT1FLTCTL is shown in Figure 32-92 and described in Table 32-90.

Return to the Summary Table.

COMPH/CEVT1 Digital filter3 Control Register

Figure 32-92. SDCOMP3EVT1FLTCTL Register

15	14	13	12	11	10	9	8
FILINIT	RESERVED	THRESH					SAMPWIN
R-0/W1S-0h	R-0h	R/W-0h					R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN				RESERVED			
R/W-0h				R-0h			

Table 32-90. SDCOMP3EVT1FLTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	High filter initialization. 0 No effect 1 Initialize all samples to the filter input value Reset type: SYSRSn
14	RESERVED	R	0h	Reserved
13-9	THRESH	R/W	0h	High filter majority voting threshold. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Reset type: SYSRSn
8-4	SAMPWIN	R/W	0h	High filter sample window size. Number of samples to monitor is SAMPWIN+1. Reset type: SYSRSn
3-0	RESERVED	R	0h	Reserved

32.12.2.79 SDCOMP3EVT1FLTCLKCTL Register (Offset = E8h) [Reset = 0000h]

SDCOMP3EVT1FLTCLKCTL is shown in Figure 32-93 and described in Table 32-91.

Return to the Summary Table.

COMPH/CEVT1 Digital filter3 Clock Control Register

Figure 32-93. SDCOMP3EVT1FLTCLKCTL Register

15	14	13	12	11	10	9	8
RESERVED						CLKPRESCALE	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
CLKPRESCALE							
R/W-0h							

Table 32-91. SDCOMP3EVT1FLTCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	CLKPRESCALE	R/W	0h	High filter sample clock prescale. Number of system clocks between samples. Reset type: SYSRSn

32.12.2.80 SDCOMP3LOCK Register (Offset = EEh) [Reset = 0000h]

SDCOMP3LOCK is shown in Figure 32-94 and described in Table 32-92.

Return to the Summary Table.

SD compartor event filter3 Lock Register

Figure 32-94. SDCOMP3LOCK Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			RESERVED	COMP	RESERVED	RESERVED	SDCOMP3CTL
R-0h			R-0h	R/WSonce-0h	R-0h	R-0h	R/WSonce-0h

Table 32-92. SDCOMP3LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	COMP	R/WSonce	0h	Lock write-access to the SDCOMP3EVT1/2FLTTCTL and COMP3FILCLKCTL registers. 0 SDCOMP3EVT1/2FLTCTL and SDCOMP3EVT1/2FLTCLKCTL registers are not locked. Write 0 to this bit has no effect. 1 SDCOMP3EVT1/2FLTCTL and SDCOMP3EVT1/2FLTCLKCTL registers are locked. Only a system reset can clear this bit. Reset type: SYSRSn
2	RESERVED	R	0h	Reserved
1	RESERVED	R	0h	Reserved
0	SDCOMP3CTL	R/WSonce	0h	Lock write-access to the SDCOMP3CTL register. 0 SDCOMP3CTL register is not locked. Write 0 to this bit has no effect. 1 SDCOMP3CTL register is locked. Only a system reset can clear this bit. Reset type: SYSRSn

32.12.2.81 SDCOMP4CTL Register (Offset = F0h) [Reset = 0000h]

SDCOMP4CTL is shown in Figure 32-95 and described in Table 32-93.

Return to the Summary Table.

SD Comparator event filter4 Control Register

Figure 32-95. SDCOMP4CTL Register

15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED		CEVT2DIGFILTSEL		RESERVED	RESERVED
R-0h	R-0h	R-0h		R/W-0h		R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED		CEVT1DIGFILTSEL		RESERVED	RESERVED
R-0h	R-0h	R-0h		R/W-0h		R-0h	R-0h

Table 32-93. SDCOMP4CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13-12	RESERVED	R	0h	Reserved
11-10	CEVT2DIGFILTSEL	R/W	0h	High comparator COMPH source select. 0 CEVT2 output drives COMPLOUT 1 Reserved 2 Output of digital filter drives COMPLOUT 3 Reserved Reset type: SYSRSn
9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5-4	RESERVED	R	0h	Reserved
3-2	CEVT1DIGFILTSEL	R/W	0h	High comparator COMPH source select. 0 CEVT1 output drives COMPHOUT 1 Reserved 2 Output of digital filter drives COMPHOUT 3 Reserved Reset type: SYSRSn
1	RESERVED	R	0h	Reserved
0	RESERVED	R	0h	Reserved

32.12.2.82 SDCOMP4EVT2FLTCTL Register (Offset = F2h) [Reset = 0000h]

SDCOMP4EVT2FLTCTL is shown in Figure 32-96 and described in Table 32-94.

Return to the Summary Table.

COMPL/CEVT2 Digital filter4 Control Register

Figure 32-96. SDCOMP4EVT2FLTCTL Register

15	14	13	12	11	10	9	8
FILINIT	RESERVED	THRESH					SAMPWIN
R-0/W1S-0h	R-0h	R/W-0h					R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN				RESERVED			
R/W-0h				R-0h			

Table 32-94. SDCOMP4EVT2FLTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	Low filter initialization. 0 No effect 1 Initialize all samples to the filter input value Reset type: SYSRSn
14	RESERVED	R	0h	Reserved
13-9	THRESH	R/W	0h	Low filter majority voting threshold. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Reset type: SYSRSn
8-4	SAMPWIN	R/W	0h	Low filter sample window size. Number of samples to monitor is SAMPWIN+1. Reset type: SYSRSn
3-0	RESERVED	R	0h	Reserved

32.12.2.83 SDCOMP4EVT2FLTCLKCTL Register (Offset = F4h) [Reset = 0000h]

SDCOMP4EVT2FLTCLKCTL is shown in Figure 32-97 and described in Table 32-95.

Return to the Summary Table.

COMPL/CEVT2 Digital filter4 Clock Control Register

Figure 32-97. SDCOMP4EVT2FLTCLKCTL Register

15	14	13	12	11	10	9	8
RESERVED						CLKPRESCALE	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
CLKPRESCALE							
R/W-0h							

Table 32-95. SDCOMP4EVT2FLTCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	CLKPRESCALE	R/W	0h	Low filter sample clock prescale. Number of system clocks between samples. Reset type: SYSRSn

32.12.2.84 SDCOMP4EVT1FLTCTL Register (Offset = F6h) [Reset = 0000h]

SDCOMP4EVT1FLTCTL is shown in Figure 32-98 and described in Table 32-96.

Return to the Summary Table.

COMPH/CEVT1 Digital filter4 Control Register

Figure 32-98. SDCOMP4EVT1FLTCTL Register

15	14	13	12	11	10	9	8
FILINIT	RESERVED	THRESH					SAMPWIN
R-0/W1S-0h	R-0h	R/W-0h					R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN				RESERVED			
R/W-0h				R-0h			

Table 32-96. SDCOMP4EVT1FLTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	High filter initialization. 0 No effect 1 Initialize all samples to the filter input value Reset type: SYSRSn
14	RESERVED	R	0h	Reserved
13-9	THRESH	R/W	0h	High filter majority voting threshold. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Reset type: SYSRSn
8-4	SAMPWIN	R/W	0h	High filter sample window size. Number of samples to monitor is SAMPWIN+1. Reset type: SYSRSn
3-0	RESERVED	R	0h	Reserved

32.12.2.85 SDCOMP4EVT1FLTCLKCTL Register (Offset = F8h) [Reset = 0000h]

SDCOMP4EVT1FLTCLKCTL is shown in Figure 32-99 and described in Table 32-97.

Return to the Summary Table.

COMPH/CEVT1 Digital filter4 Clock Control Register

Figure 32-99. SDCOMP4EVT1FLTCLKCTL Register

15	14	13	12	11	10	9	8
RESERVED						CLKPRESCALE	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
CLKPRESCALE							
R/W-0h							

Table 32-97. SDCOMP4EVT1FLTCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	CLKPRESCALE	R/W	0h	High filter sample clock prescale. Number of system clocks between samples. Reset type: SYSRSn

32.12.2.86 SDCOMP4LOCK Register (Offset = FEh) [Reset = 0000h]

SDCOMP4LOCK is shown in Figure 32-100 and described in Table 32-98.

Return to the Summary Table.

SD compartor event filter4 Lock Register

Figure 32-100. SDCOMP4LOCK Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			RESERVED	COMP	RESERVED	RESERVED	SDCOMP4CTL
R-0h			R-0h	R/WSonce-0h	R-0h	R-0h	R/WSonce-0h

Table 32-98. SDCOMP4LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	COMP	R/WSonce	0h	Lock write-access to the SDCOMP4EVT1/2FLTTCTL and COMP4FILCLKCTL registers. 0 SDCOMP4EVT1/2FLTCTL and SDCOMP4EVT1/2FLTCLKCTL registers are not locked. Write 0 to this bit has no effect. 1 SDCOMP4EVT1/2FLTCTL and SDCOMP4EVT1/2FLTCLKCTL registers are locked. Only a system reset can clear this bit. Reset type: SYSRSn
2	RESERVED	R	0h	Reserved
1	RESERVED	R	0h	Reserved
0	SDCOMP4CTL	R/WSonce	0h	Lock write-access to the SDCOMP4CTL register. 0 SDCOMP4CTL register is not locked. Write 0 to this bit has no effect. 1 SDCOMP4CTL register is locked. Only a system reset can clear this bit. Reset type: SYSRSn

Chapter 33

► COMMUNICATION PERIPHERALS

The following chapters describe the communication peripherals.

Technical Reference Manual Overview

The block diagram is shown in Figure 33-1. This Technical Reference Manual is organized into five major sections:

- C29x SYSTEM RESOURCES

  These chapters describe the C29x CPU subsystem, C29x Boot ROM, device configuration, and other system peripherals.

- ANALOG PERIPHERALS

  These chapters describe the general analog subsystem configuration, Analog-to-Digital Converter (ADC), Buffered Digital-to-Analog Converter (DAC), and Comparator Subsystem (CMPSS).

- CONTROL PERIPHERALS

  These chapters describe the Enhanced Capture (eCAP), High-Resolution Capture (HRCAP), Enhanced Pulse-Width Modulator (ePWM) with High-Resolution Pulse-Width Modulator (HRPWM), Enhanced Quadrature Encoder Pulse (eQEP), and Sigma Delta Filter Module (SDFM) peripherals.

- COMMUNICATION PERIPHERALS

  These chapters describe the communication peripherals available to the C29x subsystem such as the EtherCAT, FSI, I2C, PMBUS, UART, LIN, SPI, and SENT.

- SECURITY PERIPHERALS

  This chapter describes the safety peripherals available to the C29x subsystem such as the Hardware Security Module (HSM) and Cryptographic Accelerator.

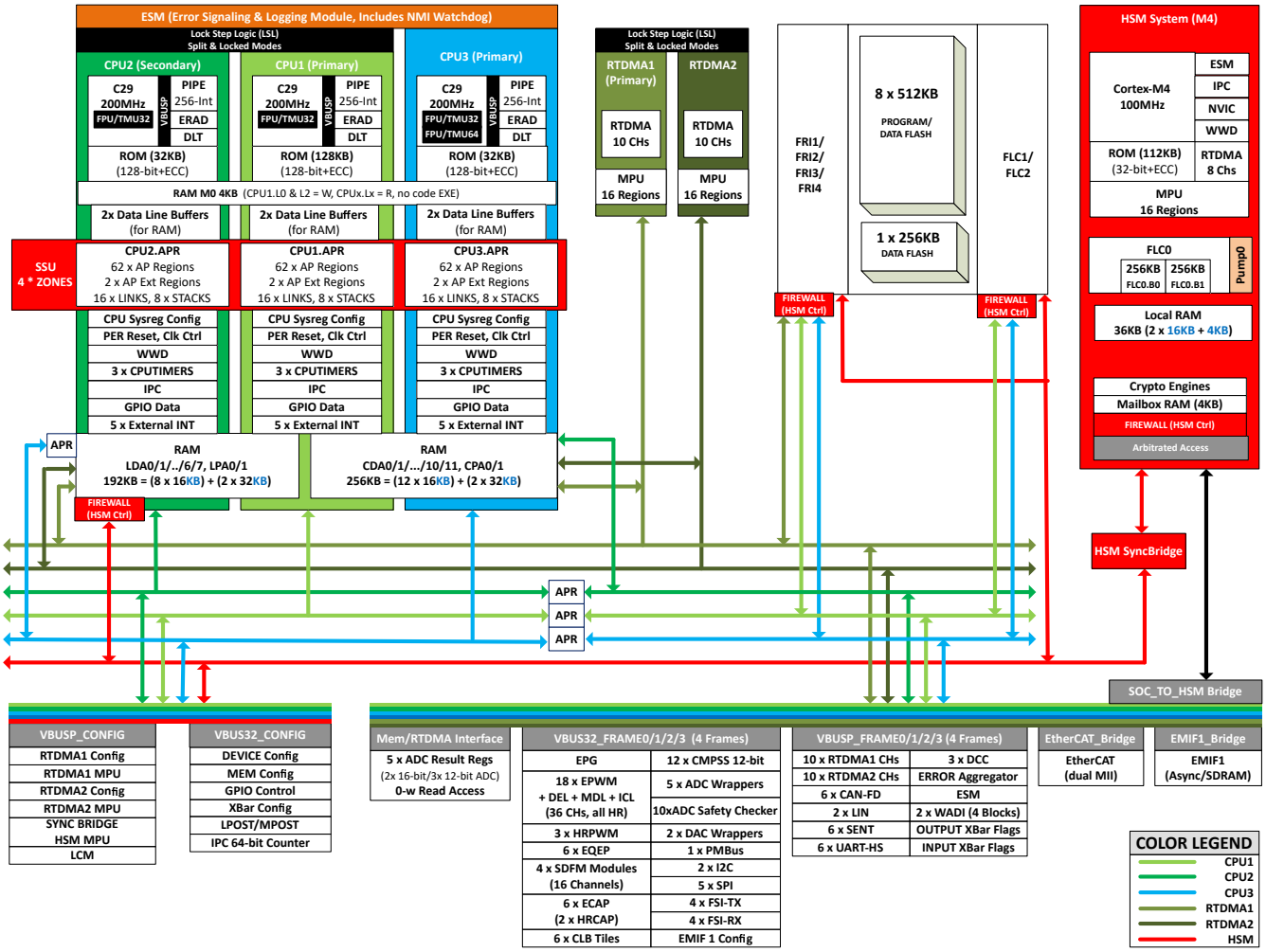


Figure 33-1. Block Diagram

Chapter 34

Modular Controller Area Network (MCAN)

This chapter describes the Modular Controller Area Network (MCAN). MCAN supports both classic CAN and CAN FD protocols.

34.1 MCAN Introduction

The Controller Area Network (CAN) is a serial communications protocol that efficiently supports distributed real-time control with a high level of reliability. CAN has high immunity to electrical interference and the ability to detect various type of errors. In CAN, many short messages are broadcast to the entire network, which provides data consistency in every node of the system.

The MCAN module supports both classic CAN and CAN FD (CAN with flexible data-rate) protocols. The CAN FD feature allows higher throughput and increased payload per data frame. Classic CAN and CAN FD devices can coexist on the same network without any conflict provided that partial network transceivers, which can detect and ignore CAN FD without generating bus errors, are used by the classic CAN devices. The MCAN module is compliant to ISO 11898-1:2015.

Note

The availability of the CAN FD feature is dependent on the device's part number. Refer to the device data sheet for more information.

Figure 34-1 shows an overview of the MCAN module.

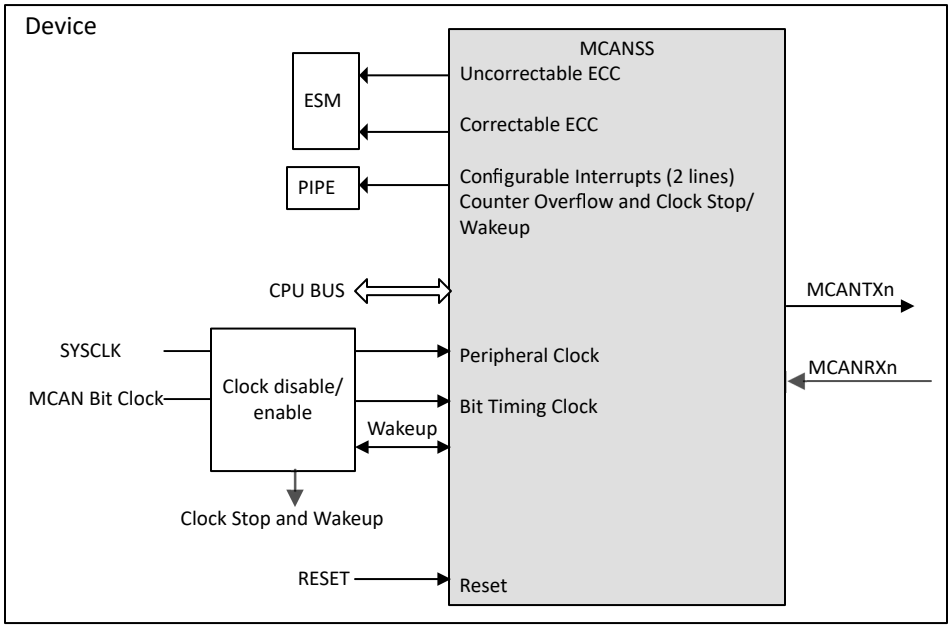


Figure 34-1. MCAN Module Overview

34.1.1 MCAN Related Collateral

Foundational Materials

- Automotive CAN Overview and Training (Video)
- C28x Academy - MCAN
- C29x Academy - CAN Flexible Data Rate (CAN-FD/MCAN)
- CAN Physical layer (Video)
- CAN and CAN FD Overview (Video)
- CAN and CAN FD Protocol (Video)

Getting Started Materials

- *Getting Started with the MCAN (CAN FD) Module Application Report*

34.1.2 MCAN Features

The MCAN module implements the following features:

- Conforms with CAN Protocol 2.0 A, B and ISO 11898-1:2015
- Full CAN FD support (up to 64 data bytes)
- AUTOSAR and SAE J1939 support
- Up to 32 dedicated transmit buffers
- Configurable transmit FIFO, up to 32 elements
- Configurable transmit queue, up to 32 elements
- Configurable transmit Event FIFO, up to 32 elements
- Up to 64 dedicated receive buffers
- Two configurable receive FIFOs, up to 64 elements each
- Up to 128 filter elements
- Loop-back mode for self-test
- Maskable interrupt (two configurable interrupt lines, correctable ECC, counter overflow, and clock stop or wakeup)
- Non-maskable interrupt (uncorrectable ECC)
- Two clock domains (CAN clock and host clock)
- ECC check for Message RAM
- Clock stop and wakeup support
- Timestamp counter
- 1-Mbps nominal bit rate, 5-Mbps data bit rate

Non-supported features:

- Host bus firewall
- Clock calibration
- Debug over CAN

34.2 MCAN Environment

The CAN network physical layer consists of a two-wire differential bus, usually twisted pair, and provides a high level of interference immunity. An external CAN transceiver IC is needed to access the bus.

Figure 34-2 shows typical MCAN wiring. Table 34-1 describes the external signals of the MCAN module.

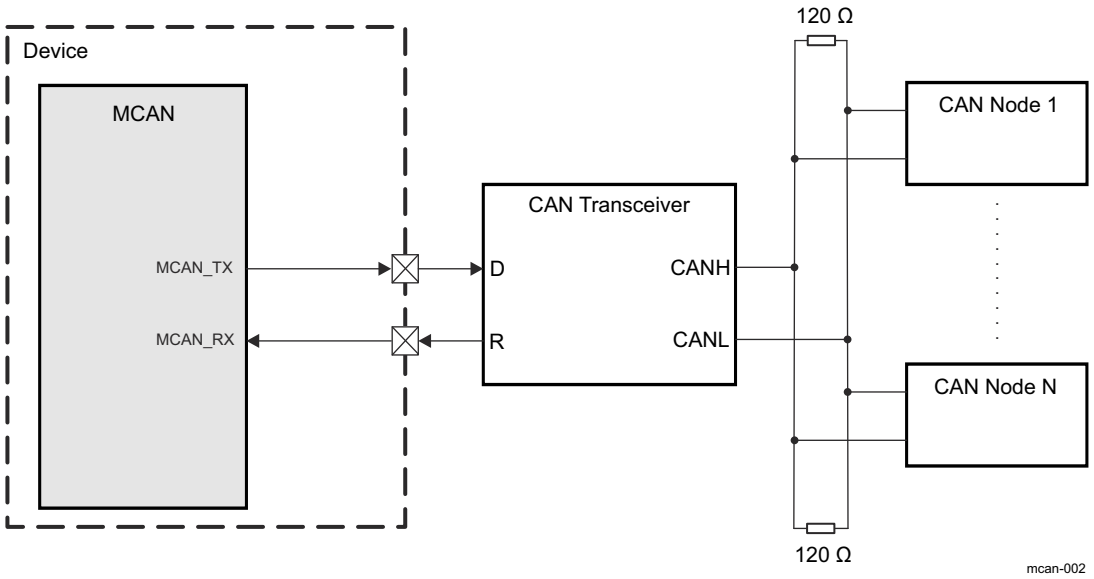


Figure 34-2. MCAN Typical Bus Wiring

Table 34-1. MCAN I/O Description

Module Signal	I/O	Description	Value at Reset
MCAN_RX	Input	Serial data input from external CAN transceiver.	HiZ
MCAN_TX	Output	Serial data output to external CAN transceiver.	HiZ

Note

See the *Terminal Configurations and Functions* section in the device data sheet and the *General-Purpose Input/Output (GPIO)* chapter to configure this peripheral to be connected to the device pins.

34.3 CAN Network Basics

The network basics are:

- The CAN bus is a 2-wire differential bus using non-return-to-zero (NRZ) encoding and has two states:
  - Recessive state (logical 1)
  - Dominant state (logical 0)
- When the bus is idle, any node can initiate a transmission to any other node (or nodes). When two or more nodes (ECUs) attempt to transmit at the same time, a non-destructive arbitration technique makes sure messages are sent in order of priority and no messages are lost.
- The message transmission is multicast. Data messages transmitted are identifier-based, not address-based.
- The content of the message is labeled by the identifier that is unique throughout the network (for example: RPM, temperature, position, pressure, and so forth).
- All nodes on the network receive the message and each performs an acceptance test on the identifier. If the message is relevant, the message is processed; otherwise, the message is ignored.
- The unique identifier also determines the priority of the message (the lower the numerical value of the identifier, the higher the priority).
- Data is transmitted and received using message frames, consisting of the following basic fields:
  - Arbitration field
  - Control field
  - Data field (up to 8 bytes for classical CAN and up to 64 bytes for CAN FD)
  - CRC field
  - ACK field

For more information, see *ISO 11898-1:2015: CAN data link layer and physical signaling*.

34.4 MCAN Integration

Figure 34-3 shows the integration of the MCAN module in the device.

Note

There can be up to six MCAN modules in this device family. The interrupt and event-trigger indices change per module instance. Refer to the *Peripheral Interrupt Priority and Expansion (PIPE)* chapter for details.

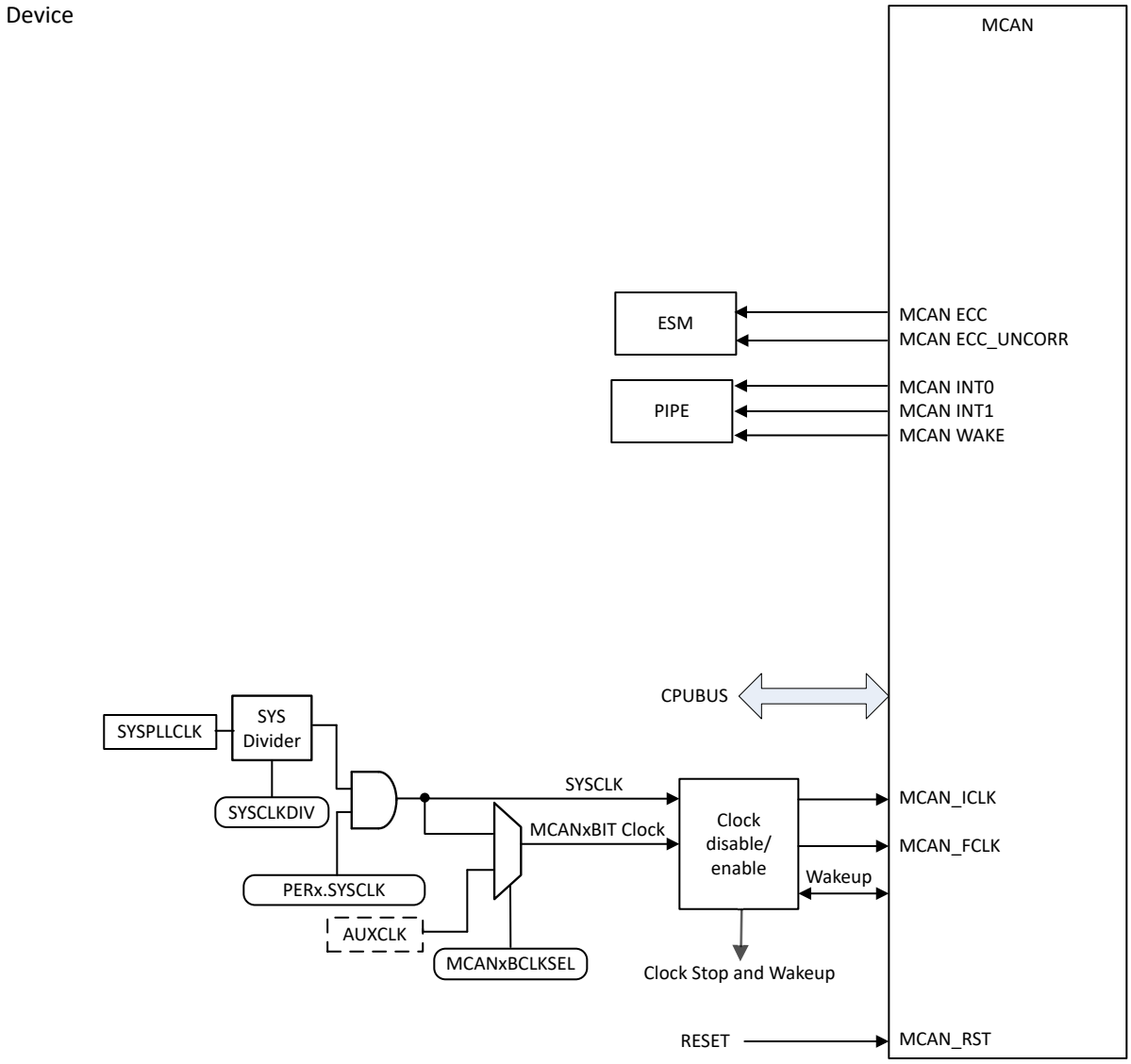


Figure 34-3. MCAN Integration

Table 34-2 summarizes the integration of the MCAN module in the device.

Table 34-2. MCAN Clocks and Resets

Destination Signal Name	Source Signal Name	Description
	Clocks	
MCAN_ICLK	SYSCLK	Interface clock for the MCAN module
MCAN_FCLK	MCANxBIT Clock	Bit timing clock for MCAN
	Resets	
MCAN_RST	RESET	Asynchronous reset signal to the MCAN module

34.5 MCAN Functional Description

The MCAN module performs CAN protocol communication according to ISO 11898-1:2015. The data bit rate can be programmed to values up to 5Mbps. Additional transceiver hardware is required for the connection to the physical layer (CAN bus).

For communication on a CAN network, individual message frames can be configured. The message frames and identifier masks are stored in the Message RAM.

All functions concerning the handling of messages are implemented in the Message Handler.

The register set of the MCAN module can be accessed directly using the module interface. These registers are used to control and configure the CAN core and the Message Handler, and to access the Message RAM.

Figure 34-4 shows the MCAN module block diagram, followed by the description of the MCAN module blocks.

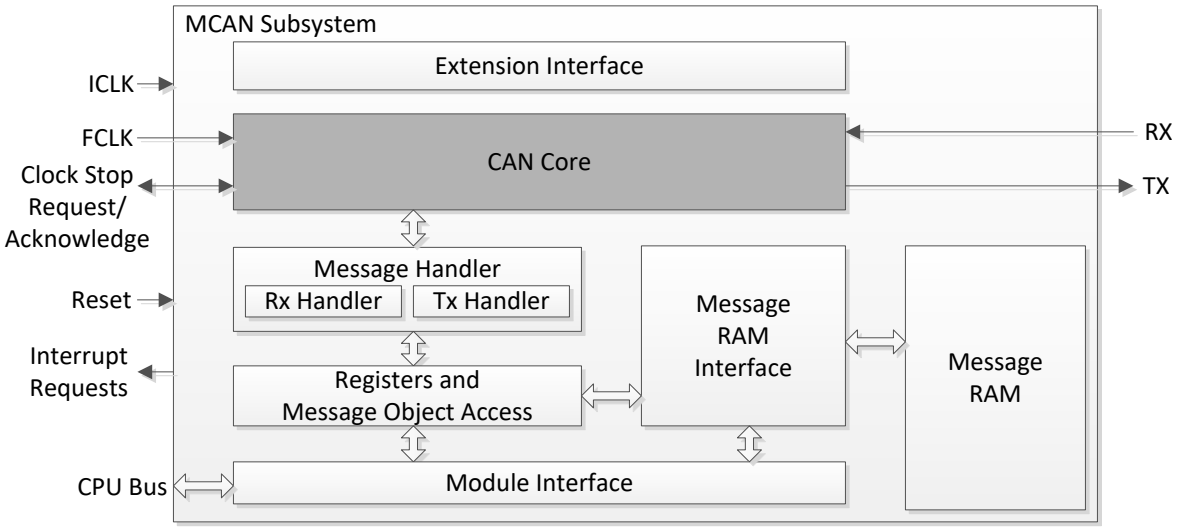


Figure 34-4. MCAN Block Diagram

- **CAN Core**: The CAN core consists of the CAN protocol controller and the Rx/Tx shift register. The CAN handles all ISO 11898-1:2015 protocol functions and supports 11-bit and 29-bit identifiers.
- **Message Handler**: the Message Handler (Rx Handler and Tx Handler) is a state machine that controls the data transfer between the single-ported Message RAM and the CAN core's Rx/Tx shift register. The Message Handler also handles the acceptance filtering and Interrupt generation as programmed in the control registers.
- **Message RAM**: the main purpose of the Message RAM is to store Rx/Tx messages, Tx Event elements, and Message ID Filter elements (for more information, see Section 34.5.16).
- **Message RAM Interface**: enables a connection between the Message RAM and the other blocks in the MCAN module.
- **Registers and Message Object Access**: Data consistency is provided by indirect accesses to the message objects. During normal operation, all software and RTDMA accessesDMA accesses to the Message RAM are done through interface registers. The interface registers have the same word-length as the Message RAM.
- **Module Interface**: The MCAN module registers are accessed by the user's software through a 32-bit peripheral bus interface.
- **Clocking**: Two clocks are provided to the MCAN module: the peripheral synchronous clock (interface clock - MCAN_ICLK) and the peripheral asynchronous clock (functional clock - MCAN_FCLK).
- **Extension Interface**: All selected internal status and control signals are routed to this interface (except for the indication signals of configuration change enable bit (MCAN_CCCR.CCE) and Interrupt Register bits (MCAN_IR).

34.5.1 Module Clocking Requirements

Two clocks are provided to the MCAN module:

- Host Clock : peripheral synchronous clock (MCAN_ICLK) as the general module clock source, and
- CAN Clock: peripheral asynchronous clock (MCAN_FCLK) provided to the CAN core for generating the CAN bit timing.

Within the MCAN module, there is a synchronization mechanism implemented to make sure there is safe data transfer between the two clock domains. There is synchronization between the signals from the Host clock domain to the CAN clock domain and conversely, and between the reset signal (MCAN_RST) to the Host clock domain and to the CAN clock domain.

Note

MCAN_ICLK must always be higher or equal to MCAN_FCLK, to achieve a stable functionality of the MCAN module: $f_{ICLK} \geq f_{FCLK}$

The CAN-FD supports higher speeds of operation and as such has more stringent timing requirements than the classic CAN. For performance, TI recommends using the lowest N-divider value that maintains a working PLL REF_CLK for the system. Lower N-divider values increase the loop bandwidth of the PLL, which in turn improves timing margins for CAN-FD.

34.5.2 Interrupt Requests

The MCAN module generates interrupt requests and is configured using the Host CPU. The Suspend mode prevents the interrupt requests from propagating to the Host CPU. The MCAN core has two interrupt lines and 30 internal interrupt sources. Each source can be configured to drive one of the two interrupt lines. The interrupts are level high interrupts. The MCAN core provides two interrupt requests (MCANSS_INT0 and MCANSS_INT1).

For more information, see the following registers:

- Interrupt Register (MCAN_IR)
- Interrupt Enable (MCAN_IE)
- Interrupt Line Select (MCAN_ILS)
- Interrupt Line Enable (MCAN_ILE)

The MCAN module supports External Timestamp Counter. The External Timestamp Counter produces an interrupt when the count rolls over (see Section 34.5.10.1).

For more information, see the following registers:

- Interrupt Clear Shadow Register (MCANSS_ICS)
- Interrupt Raw Status Register (MCANSS_IRS)
- Interrupt Enable Clear Shadow Register (MCANSS_IECS)
- Interrupt Enable Register (MCANSS_IE)
- Interrupt Enable Status Register (MCANSS_IES)
- End Of Interrupt Register (MCANSS_EOI)
- External Timestamp Prescaler Register (MCANSS_EXT_TS_PRESCALER)
- External Timestamp Unserviced Interrupts Counter Register (MCANSS_EXT_TS_UNSERVICED_INTR_CNTR)

To clear IRQ_INT0, IRQ_INT1, and TS_WAKE interrupts, write to the EOI bit field for the corresponding interrupt number that is described in the MCANSS_EOI register. When the MCAN is used by the CPU, in addition to clearing the interrupt sources, additional operations are warranted in the PIPE module (refer to the *Configuring Interrupts* section of the *Peripheral Interrupts Priority and expansion* chapter) for successive MCAN interrupts to be recognized.

The MCAN module is capable of issuing an ECC interrupt. After clearing the ECC interrupt source, the application software must also write a 1 to the EOI registers (MCANERR_SEC_EOI.EOI_WR/ MCANERR_DED_EOI.EOI_WR). For more information, see Section 34.5.12.2.

34.5.3 Operating Modes

The operating modes are discussed in the following sections.

34.5.3.1 Software Initialization

A software initialization begins when the MCAN_CCCR.INIT bit is set to 1. This is done either by software or by a hardware reset, when an uncorrected bit error is detected in the Message RAM, or by going to a Bus_Off state. While the MCAN_CCCR.INIT bit is set, the message transfer is stopped and the status of the output TX pin is recessive (high). The counters of the Error Management Logic (EML) are unchanged. Setting the MCAN_CCCR.INIT bit does not change any configuration register. Resetting the MCAN_CCCR.INIT bit finishes the software initialization. After waiting for the occurrence of a sequence of 11 consecutive recessive bits (indication for Bus_Idle state) the message transfer starts.

Access to the MCAN configuration registers is only enabled when both MCAN_CCCR.INIT and MCAN_CCCR.CCE bits are set (write protection).

The MCAN_CCCR.CCE bit can only be set/reset while the MCAN_CCCR.INIT = 1. The MCAN_CCCR.CCE bit is automatically reset when the MCAN_CCCR.INIT bit is reset.

The following registers are reset when the MCAN_CCCR.CCE bit is set:

- MCAN_HPMS - High Priority Message Status
- MCAN_RXF0S - Rx FIFO 0 Status
- MCAN_RFX1S - Rx FIFO 1 Status
- MCAN_TXFQS - Tx FIFO/Queue Status
- MCAN_TXBRP - Tx Buffer Request Pending
- MCAN_TXBTO - Tx Buffer Transmission Occurred
- MCAN_TXBCF - Tx Buffer Cancellation Finished
- MCAN_TXEFS - Tx Event FIFO Status

The Timeout Counter value MCAN_TOCV.TOC field is preset to the value configured by the MCAN_TOCC.TOP field when the MCAN_CCCR.CCE bit is set.

In addition, the Tx Handler and Rx Handler are held in idle state while MCAN_CCCR.CCE = 1.

The following registers are only writable while MCAN_CCCR.CCE = 0

- MCAN_TXBAR - Tx Buffer Add Request
- MCAN_TXBCR - Tx Buffer Cancellation Request

MCAN_CCCR.TEST and MCAN_CCCR.MON bits can only be set by the Host CPU while MCAN_CCCR.INIT = 1 and MCAN_CCCR.CCE = 1. Both bits are reset at any time. The MCAN_CCCR.DAR bit can only be set/reset while MCAN_CCCR.INIT = 1 and MCAN_CCCR.CCE = 1.

Table 34-3 shows the steps to configure the MCAN module.

Table 34-3. Steps to Configure MCAN Module

Step	Operation	Description	Pseudo Code
1	Initialize MCAN_CCCR	Set MCAN_CCCR.INIT bit and check that the bit has been set	INIT = 1; If INIT ≠ 1, wait until set
2	Unlock protected registers	Set MCAN_CCCR.CCE bit	CCE = 1;
3	Configure CAN mode	Set MCAN_CCCR.FDOE bit to CAN FD	FDOE = 1 for CAN FD FDOE = 0 for Classic CAN
4	Configure Bit Rate Switching	Set MCAN_CCCR.BRSE bit	BRSE = 1 for bit rate switching BRSE = 0 for no bit rate switching
5	Set nominal bit timing(1)	Set MCAN_NBTP register	
6	Lock protected registers	Clear MCAN_CCCR.CCE bit	CCE = 0;

Table 34-3. Steps to Configure MCAN Module (continued)

Step	Operation	Description	Pseudo Code
7	Return MCAN module to normal operation	Clear MCAN_CCCR.INIT bit and check that the bit has been cleared	INIT = 0; If INIT ≠ 0, wait until cleared

(1) See the MCAN_NBTP register on how to program CAN bit timing in the *MCAN_REGS Registers* section.

34.5.3.2 Normal Operation

Once the MCAN module is initialized and the MCAN_CCCR.INIT bit is reset to zero, the MCAN module synchronizes to the CAN bus and is ready for communication. After passing the acceptance filtering, received messages including Message Identifier (ID) and Data Length Code (DLC) are stored into a dedicated Rx Buffer or into Rx FIFO 0/Rx FIFO 1.

For messages to be transmitted, dedicated Tx buffers, and a Tx FIFO or a Tx queue can be initialized or updated.

Note

The automated transmission upon reception of remote frames is not supported.

34.5.3.3 CAN FD Operation

The CAN FD standard allows extended frames to be sent, up to 64 data bytes in a single frame at a higher bit rate for the data phase of a frame, up to 5Mbps. The CAN FD standard introduces the ability to switch from one bit rate to another. Extended Data Length (EDL), as shown in Figure 34-5 and described in Table 34-4, sets a data length of up to 8 or 64 data bytes. Bit Rate Switching (BRS) indicates whether two bit rates (the data phase is transmitted at a different bit rate compared to the arbitration phase) are enabled.

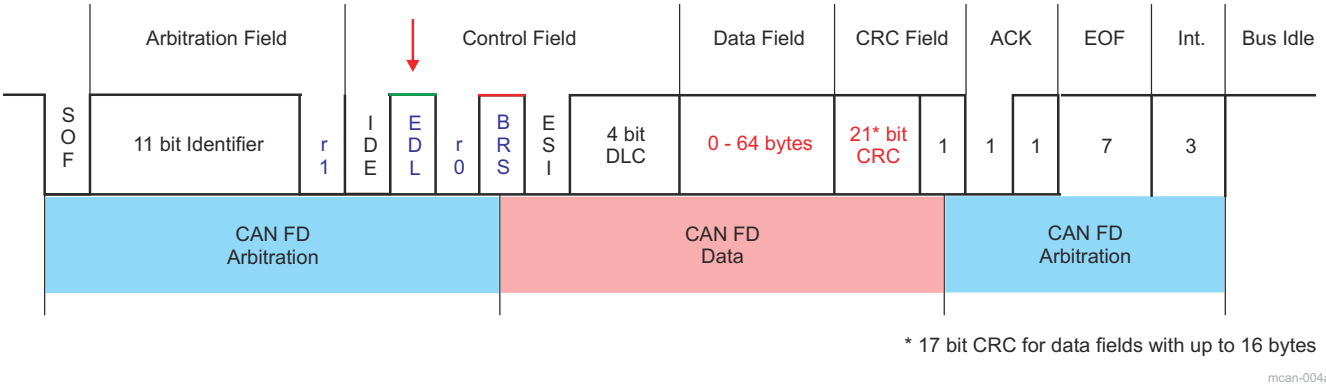


Figure 34-5. CAN FD Frame

Table 34-4. CAN FD Frame Description

Bit	Description
SOF	Start of Frame
IDE	Identifier extension (for 29 bit extended ID)
FDF	Flexible Data Format
BRS	Bit Rate Switching
ESI	Error Status Indicator
DLC	Data Length Code
CRC	Cyclic Redundancy check

There are two variants of CAN FD frame transmission:

- CAN FD frame transmission without bit rate switching
- CAN FD frame transmission where control field, data field, and CRC field are transmitted with a higher bit rate than the beginning and the end of the frame

The previously reserved bit in CAN frames with 11-bit identifiers and the first previously reserved bit in CAN frames with 29-bit identifiers are now decoded as the FDF bit. In the CAN frames, FDF = recessive (logical 1) signifies a CAN FD frame, FDF = dominant (logical 0) signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF - res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive. Note that the coding of res = recessive is reserved for future expansion of the protocol. If the MCAN module receives a frame with FDF = recessive and res = recessive, the MCAN signals a Protocol Exception Event by setting the MCAN_PSR.PXE bit. When Protocol Exception Handling is enabled (MCAN_CCCR.PXHD = 0), this causes the operation state to change from Receiver (MCAN_PSR.ACT = 10) to Integrating (MCAN_PSR.ACT = 00) at the next sample point. In case Protocol Exception Handling is disabled (MCAN_CCCR.PXHD = 1), the MCAN treats a recessive bit as an error and responds with an error frame.

CAN FD operation is enabled by programming the MCAN_CCCR.FDOE bit. If MCAN_CCCR.FDOE = 1, transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured using the FDF bit in the respective Tx Buffer element.

With MCAN_CCCR.FDOE = 0, received frames are interpreted as Classic CAN frames, which leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if the FDF bit of a Tx Buffer element is set. The MCAN_CCCR.FDOE and MCAN_CCCR.BRSE bits can only be changed while the MCAN_CCCR.INIT and MCAN_CCCR.CCE bits are both set. With MCAN_CCCR.FDOE = 0, the setting of bits FDF and BRS is ignored and frames are transmitted in Classic CAN format.

With MCAN_CCCR.FDOE = 1 and MCAN_CCCR.BRSE = 0, only FDF bit of a Tx Buffer element is evaluated. With MCAN_CCCR.FDOE = 1 and MCAN_CCCR.BRSE = 1, transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with bits FDF and BRS set are transmitted in CAN FD format with bit rate switching.

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significantly higher than in the CAN FD arbitration phase. In this case, disable the CAN FD bit rate switching option for transmissions.
- During system startup, all nodes are transmitting Classic CAN messages until verified that the nodes are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wakeup messages in CAN Partial Networking must be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD capable. Non CAN FD nodes are held in Silent mode until programming has completed. Then all nodes switch back to Classic CAN communication.

The coding of the DLC in the CAN FD format differs from the standard CAN format. The DLC codes 0 to 8 have the same coding as in standard CAN (0 to 8 data bytes), the codes 9 to 15, which in standard CAN all code a data field of 8 bytes, are coded according to Table 34-5.

Table 34-5. DLC Coding in CAN FD

DLC	9	10	11	12	13	14	15
Number of Data Bytes	12	16	20	24	32	48	64

For CAN FD frames, the bit timing is switched inside the frame after the BRS (Bit Rate Switch) bit in case this bit is recessive. In the CAN FD arbitration phase, before the BRS bit, the nominal CAN bit timing (see Figure 34-6) is used as configured by the Nominal Bit Timing and Prescaler Register (MCAN_NBTP). In the following CAN FD data phase, the data phase bit timing is used as configured by the Data Bit Timing and Prescaler Register

(MCAN_DBTP). The bit timing is switched back from the data phase timing at the CRC delimiter or when an error is detected, whichever occurs first.

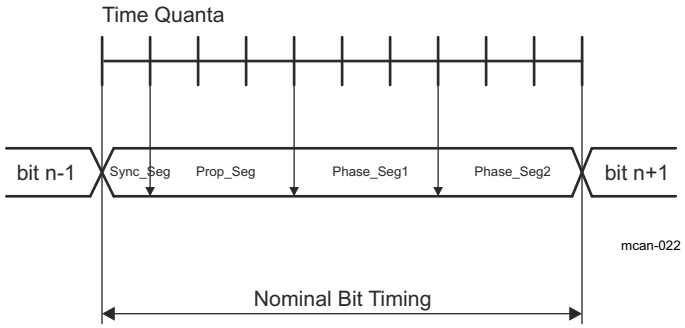


Figure 34-6. CAN Bit Timing

The maximum configurable data phase bit timing depends on the CAN clock frequency (MCAN_FCLK). Example: with MCAN_FCLK = 20MHz and the shortest configurable bit time of 4 t_q (time quanta), the bit rate in the data phase is 5Mbit/s.

In both data frame formats, CAN FD and CAN FD with bit rate switching, the value of the Error Status Indicator (ESI) bit depends on the transmitter error state (see MCAN_PSR.RESI bit) monitored at the start of the transmission. If the transmitter has an error passive flag, the ESI bit is transmitted recessive; else, the ESI bit is transmitted dominant.

34.5.4 Transmitter Delay Compensation

34.5.4.1 Description

When only one CAN FD node is transmitting and all other nodes are receivers, the length of the bus line has no impact. When transmitting using the TX pin, the MCAN module receives the transmitted data from the CAN transceiver using the RX pin. The received data is delayed. If the transmitter delay is greater than TSEG1 (time segment before sample point), a bit error is detected.

The MCAN module provides a delay compensation mechanism to compensate for the transmitter delay. The compensation mechanism enables transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver. Without transmitter delay compensation the bit rate in the data phase is limited by the transmitter delay.

The mechanism enables configurations where the data bit time is shorter than the transmitter delay (it is described in detail in ISO 11898-1:2015). The transmitter delay compensation is enabled by setting the MCAN_DBTP.TDC bit to 1.

The delayed transmit data is compared against the received data at the Secondary Sample Point (SSP) to check for bit errors during the data phase of transmitting nodes. If a bit error is detected, the transmitter reacts on this bit error at the next following regular sample point. During the arbitration phase, the delay compensation is always disabled.

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the MCAN's transmit output TX pin through the transceiver to the receive input RX pin plus the transmitter delay compensation offset configured by the MCAN_TDCR.TDCO field (see Figure 34-7). The transmitter delay compensation offset is used to adjust the position of the SSP inside the received bit (example: half of the bit time in the data phase). The position of the SSP is rounded down to the next integer number of mtq.

The actual transmitter delay compensation value can be checked by reading the MCAN_PSR.TDCV field. This field is cleared when the MCAN_CCCR_INIT bit is set and is updated at each transmission of CAN FD frame while the MCAN_DBTP.TDC bit is set.

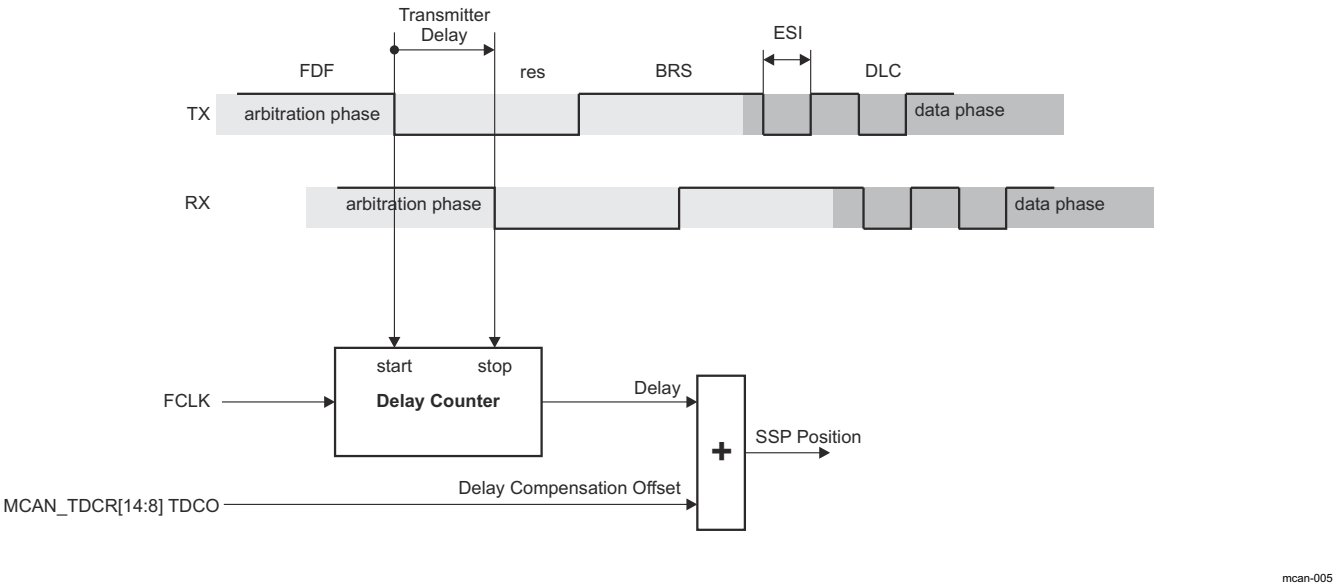


Figure 34-7. Transmitter Delay Measurement

34.5.4.2 Transmitter Delay Compensation Measurement

When transmitter delay compensation is enabled (by programming MCAN_DBTP.TDC = 1), the measurement is started within each transmitted CAN FD frame at the falling edge of FDF bit to bit r0. The measurement is stopped when this edge is seen at the receive input RX pin of the transmitter. The resolution of this measurement is one mtq (see Figure 34-7). The mtq (minimum time quantum) dimension is equal to the CAN clock period (MCAN_FCLK).

The use of a transmitter delay compensation filter window can be enabled by programming the MCAN_TDCR.TDCF field. This filter feature defines a minimum value for the SSP position to avoid the case in which a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in an early taken SSP position. Dominant edges on the RX pin that result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least MCAN_TDCR.TDCF field and the RX pin is low.

The following boundary conditions must be considered:

- The sum of the measured delay from the TX pin to the RX pin and the configured transmitter delay compensation offset (MCAN_TDCR.TDCO field) is less than 6 bit-times in the data phase.
- The sum of the measured delay from the TX pin to the RX pin and the configured transmitter delay compensation offset (MCAN_TDCR.TDCO) field is less than or equal to 127 mtq. In case this sum exceeds 127 mtq, the maximum value of 127 mtq is used for transmitter delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of receive bits at the SSPs.

34.5.5 Restricted Operation Mode

In restricted operation mode, the CAN node is able to receive data and remote frames and to give acknowledgment to valid frames, but the node does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, the node does not send dominant bits; instead the node waits for the occurrence of bus idle condition to resynchronize to the CAN communication. The receive and transmit error counters (MCAN_ECR.REC and MCAN_ECR.TEC) are frozen while CAN error logging (MCAN_ECR.CEL) is active. The Host CPU can set the MCAN module into Restricted Operation Mode by setting the MCAN_CCCR.ASM bit. The bit can only be set by the Host CPU at any time when both MCAN_CCCR.CCE and MCAN_CCCR.INIT bits are set to 1.

The restricted operation mode is automatically entered when the Tx Handler is not able to read data from the Message RAM in time. To leave restricted operation mode, the Host CPU has to reset the MCAN_CCCR.ASM bit. This mode can be used in applications that adapt themselves to different CAN bit rates. In this case, the application tests different bit rates and leaves the restricted operation mode after the node has received a valid frame.

Note

The Restricted Operation Mode must not be combined with the Loop Back Mode.

34.5.6 Bus Monitoring Mode

Entering bus monitoring mode is done by setting the MCAN_CCCR.MON bit to 1. In this mode (see ISO 11898-1:2015, *Bus Monitoring* section), the MCAN module is able to receive valid data and remote frames, but cannot start a transmission. The MCAN module sends only recessive bits on the CAN bus. If the MCAN module is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the MCAN module monitors this dominant bit, although the CAN bus can remain in recessive state. In bus monitoring mode, the MCAN_TXBRP register is held in reset state. The bus monitoring mode can be used to analyze the traffic on a CAN bus without affecting the bus by the transmission of dominant bits. Figure 34-8 shows the connection of the TX and RX signals to the MCAN module in bus monitoring mode.

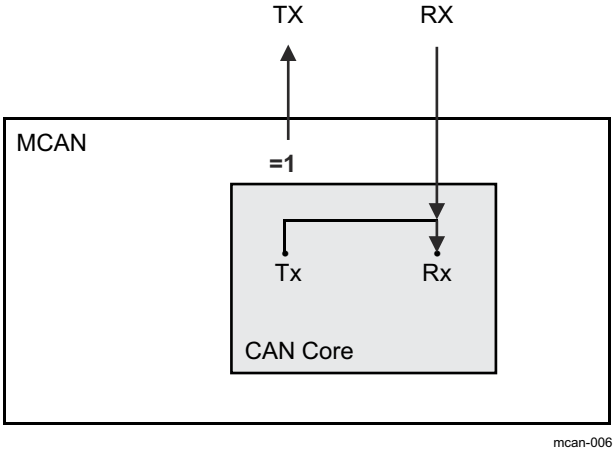


Figure 34-8. Connection of Signals in Bus Monitoring Mode

34.5.7 Disabled Automatic Retransmission (DAR) Mode

According to the CAN Specification (see ISO11898-1:2015, *Recovery Management* section), the MCAN module provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled (see the MCAN_CCCR.DAR bit).

34.5.7.1 Frame Transmission in DAR Mode

In DAR mode, the automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission is disabled. A Tx buffer's Tx Request Pending (MCAN_TXBRP[xx]) TRPx bit is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

Successful transmission:

- Corresponding Tx Buffer Transmission Occurred (MCAN_TXBTO[xx]) TOx bit is set
- Corresponding Tx Buffer Cancellation Finished (MCAN_TXBCF[xx]) CFx bit is not set

Successful transmission in spite of cancellation:

- Corresponding Tx Buffer Transmission Occurred (MCAN_TXBTO[xx]) TOx bit is set
- Corresponding Tx Buffer Cancellation Finished (MCAN_TXBCF[xx]) CFx bit is set

Arbitration lost or frame transmission disturbed:

- Corresponding Tx Buffer Transmission Occurred (MCAN_TXBTO[xx]) TOx bit is not set
- Corresponding Tx Buffer Cancellation Finished (MCAN_TXBCF[xx]) CFx bit is set

In the case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = 10 (transmission in spite of cancellation).

34.5.8 Clock Stop Mode

Entering clock stop mode is controlled by the input clock stop request signal or MCAN_CCCR.CSR bit. As long as the clock stop request signal is active, the MCAN_CCCR.CSR bit is read as 1. When all pending transmission requests have completed, the MCAN module waits until bus idle state is detected. Then the MCAN module sets the MCAN_CCCR.INIT to 1 to prevent any further CAN transfers. The MCAN module acknowledges that the module is ready for power down by setting the output clock stop acknowledge signal to 1 and the MCAN_CCCR.CSA bit to 1. In this state, before the clocks are switched off, further register accesses can be made. A write access to the MCAN_CCCR.INIT bit has no effect. Now the module clock inputs MCAN_ICLK and MCAN_FCLK can be switched off.

To leave power-down mode, the application has to turn on the module clocks before resetting the input clock stop request signal respectively the MCAN_CCCR.CSR flag bit. The MCAN acknowledges this by resetting the output clock stop acknowledge signal respectively the MCAN_CCCR.CSA flag bit. Afterwards, the application can restart CAN communication by resetting the MCAN_CCCR.INIT bit.

Restoring the clocks from clock stop mode needs to be done according to how the clock stop was initiated.

The MCAN module supports two external clock stop modes:

- Immediate
- Graceful

In a graceful clock stop mode when the clock stop request is asserted, the MCAN core responds with a clock stop acknowledge when all pending Tx messages have been processed and an Idle line had been detected. The MCAN_CCCR.INIT bit is set, the MCAN core goes and stays Idle.

The automatic wakeup feature is enabled by setting the MCANSS_CTRL.AUTOWAKEUP and MCANSS_CTRL.WAKEUPREQEN bits to 1 (for more information, see Section 34.5.8.2). When an external clock stop request is removed and no suspend request is active, a read-modify-write to the MCAN_CCCR.INIT bit is performed to clear the bit.

34.5.8.1 Suspend Mode

The MCAN module supports two suspend modes:

- Immediate
- Graceful

In a graceful suspend mode (see the MCANSS_CTRL.DBGSUSP_FREE bit) when the suspend request is asserted, a clock stop request to the MCAN core is performed. The MCAN core responds with a clock stop acknowledge when all pending Tx messages have been processed and an Idle line had been detected. At that point, the MCAN_CCCR.INIT bit is set and the MCAN core stays Idle. The suspend state can be verified by reading the MCAN_CCCR.INIT bit.

The automatic wakeup feature is enabled by setting the MCANSS_CTRL.AUTOWAKEUP and MCANSS_CTRL.WAKEUPREQEN bits to 1 (for more information, see Section 34.5.8.2). When suspend request is removed, if no external clock stop request is active, a read-modify-write to the MCAN_CCCR.INIT bit is performed to clear the bit.

During suspend mode the auto-clear feature is disabled. The following register fields have an auto-clear feature:

- MCAN_ECR.CEL
- MCAN_PSR.LEC
- MCAN_PSR.DLEC
- MCAN_PSR.RESI
- MCAN_PSR.RBRS
- MCAN_PSR.RFDF
- MCAN_PSR.PXE

34.5.8.2 Wakeup Request

Issuing a clock stop request puts the MCAN module into power-down mode (Sleep Mode). During transition from IDLE to ACTIVE, if the MCANSS_CTRL.AUTOWAKEUP and MCANSS_CTRL.WAKEUPREQEN bits are enabled, after the MCAN Core responds to the removal of the clock stop request with removing the clock stop acknowledge, a read-modify-write is issued to clear the MCAN_CCCR.INIT bit and the MCAN core resumes operation.

If the MCANSS_CTRL.WAKEUPREQEN bit is set, the MCAN module provides a wakeup request on the following wakeup event:

- The receive RX pin is dominant (logical 0)

The wakeup request is deasserted when any of the following conditions occur:

- Clock stop request is removed and clock stop acknowledge is deasserted
- A reset is applied to the MCAN module

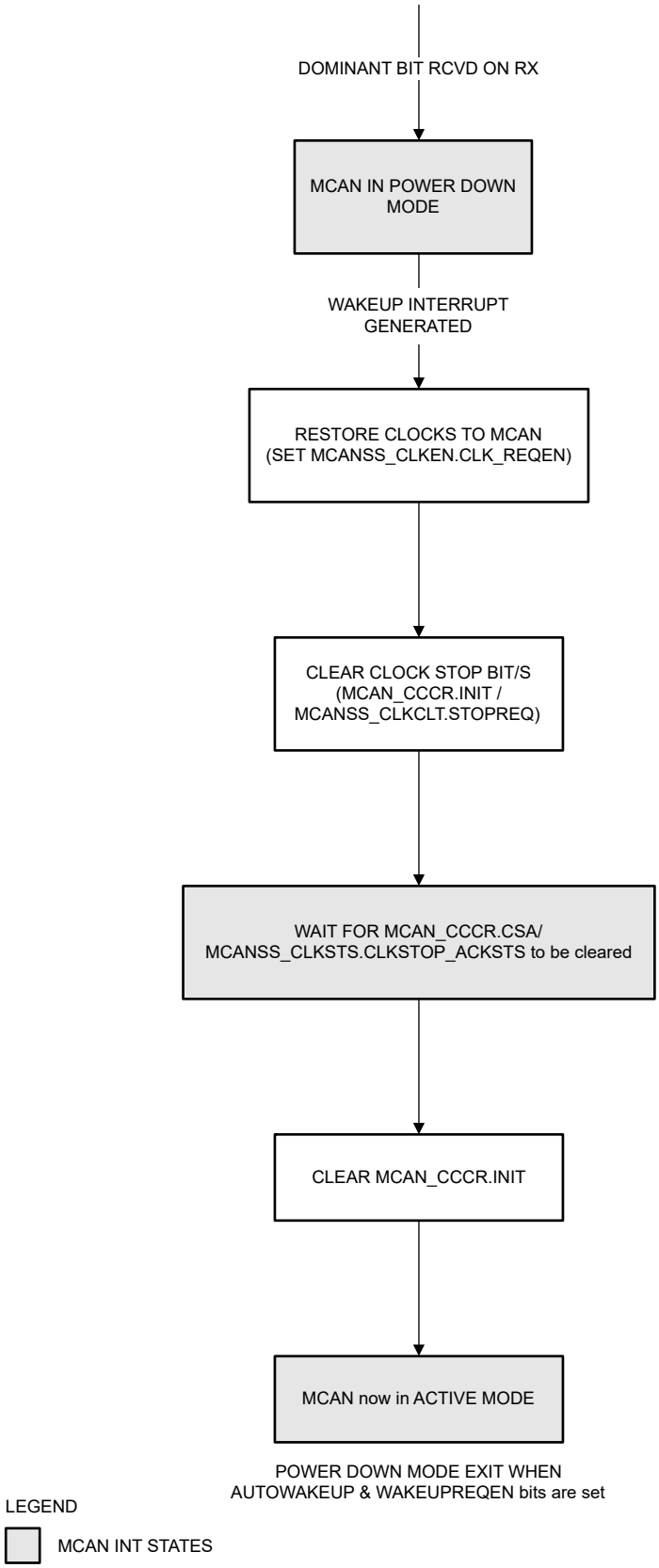


Figure 34-9. Auto Wakeup Enabled Exit from Power Down

34.5.9 Test Modes

The MCAN_TEST register write access is enabled by setting the test mode enable MCAN_CCCR.TEST bit to 1. The MCAN_TEST register allows the configuration of the test modes and test functions.

The transmit (TX) pin has four different output functions which can be selected by programming the MCAN_TEST.TX field. The default function is the serial data output. The pin can also be driven with a constant dominant or recessive value. It is also possible to drive the sample-point signal to monitor the bit-timing.

The actual value of the receive (RX) pin can be monitored from MCAN_TEST.RX bit. Both functions can be used to check the physical layer. Due to the synchronization mechanism between the CAN clock (MCANx_FCLK) and Host clock (MCANx_ICLK) domain, there can be a delay of several Host clock periods between writing to the MCAN_TEST.TX field until the new configuration is visible at the output TX pin. This applies also when reading input RX pin by way of the MCAN_TEST.RX bit.

Note

Test modes can be used for self-test only. The software control for TX pin interferes with all CAN protocol functions. It is not recommended to use test modes for an application.

34.5.9.1 External Loop Back Mode

The MCAN module can be set into external loop back mode by programming MCAN_TEST.LBCK to 1. In loop back mode, the MCAN treats the transmitted messages as received messages and stores the messages (if the messages pass acceptance filtering) into an Rx Buffer or an Rx FIFO. Figure 34-10 shows the connection of the TX and RX pins to the MCAN module in external loop back mode.

This mode is provided for hardware self-test. To be independent from external stimulation, the MCAN module ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remote frame) in loop back mode. In this mode, the MCAN module performs an internal feedback from the Tx output to the Rx input. The actual value of the RX input pin is disregarded by the MCAN module. The transmitted messages are monitored at the TX pin.

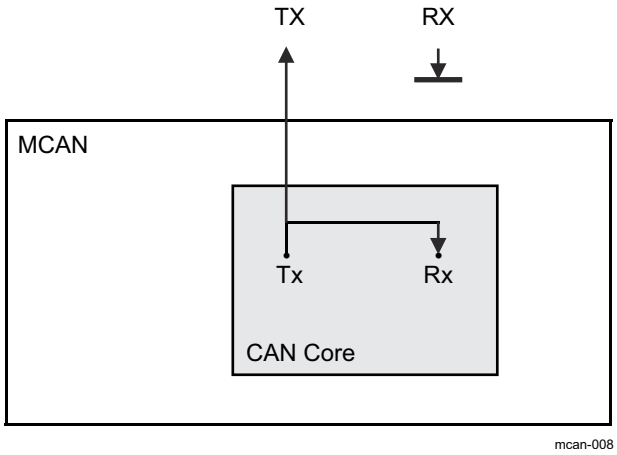


Figure 34-10. External Loop Back Mode

34.5.9.2 Internal Loop Back Mode

The MCAN module can be set into internal loop back mode by programming MCAN_TEST.LBCK and MCAN_CCCR.MON bits to 1. The internal loop back mode is used for a Hot Self-test. The Hot Self-test allows the MCAN module to be tested without affecting a running CAN system connected to the TX and RX pins. In this mode, the RX pin is disconnected from the MCAN module and the TX pin is held recessive. Figure 34-11 shows the connection of the TX and RX pins to the MCAN module in internal loop back mode.

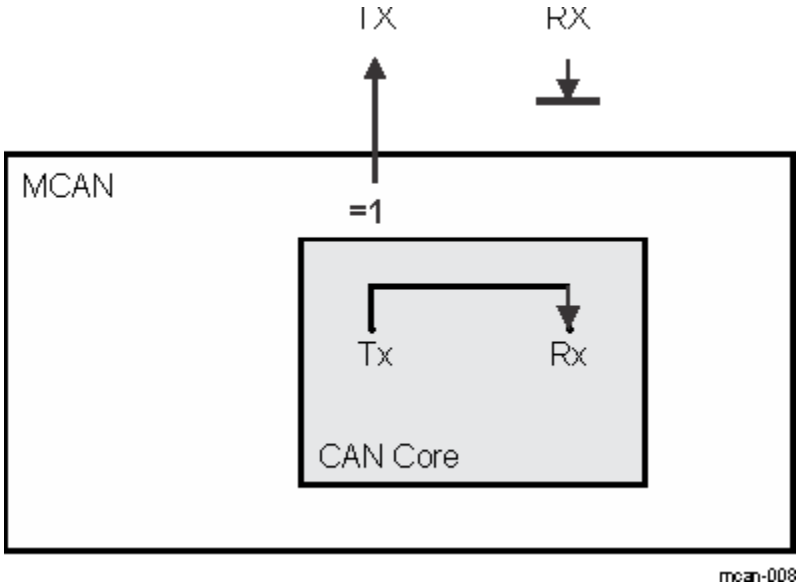


Figure 34-11. Internal Loop Back Mode

34.5.10 Timestamp Generation

The MCAN module has integrated a 16-bit wrap-around counter for timestamp generation. The timestamp counter prescaler MCAN_TSCC.TCP field can be configured to clock the counter in multiples of CAN bit times (1-16). The counter is readable by way of the MCAN_TSCV.TSC field. A write access to the MCAN_TSCV register resets the counter to zero. When the timestamp counter wraps around the interrupt MCAN_IR.TSW flag is set. On start of a frame reception/transmission the counter value is captured and stored into the timestamp section of an Rx Buffer/Rx FIFO (RXTS[15:0]) or Tx Event FIFO (TXTS[15:0]) element. For more information, see Section 34.5.16.

34.5.10.1 External Timestamp Counter

For CAN FD operation mode, the MCAN core requires an external timestamp counter (see Figure 34-12). An externally generated 16-bit vector can substitute the integrated 16-bit CAN bit time counter (internal timestamp counter) for receive and transmit timestamp generation. An external 16-bit timestamp counter can be used by programming the MCAN_TSCC.TSS field.

The external timestamp counter uses the interface clock (MCANx_ICLK) as a reference clock. The MCAN core accepts a 16-bit timestamp. A 24-bit prescaler provides a programmable resolution for the timestamp (see MCANSS_EXT_TS_PRESCALER.PRESCALER bit field). The external timestamp counter can be enabled or disabled through the MCANSS_CTRL.EXT_TS_CNTR_EN bit. When disabled, the counter is reset back to zero. While enabled, the counter keeps incrementing. When the timestamp rolls over, the MCAN_IRQ_TS interrupt is generated.

When the timestamp rolls over, the MCANSS_IRS register is set. The MCANSS_IE register can be affected by writing to the MCANSS_IESS register to set or to the MCANSS_IECS register to clear. The MCANSS_IESS register is a shadow register mapped to the same address as the MCANSS_IE register. The level interrupt is a reflection of both MCANSS_IRS and MCANSS_IE being set. The MCANSS_IES register reflects the level interrupt. When a rollover event occurs, the interrupt counter is incremented. Writing to the MCANSS_ICS register to clear the MCANSS_IRS register also decrements the interrupt counter. Writing to the MCANSS_EOI register issues another pulse, if the interrupt counter is not zero.

The rollover event can be artificially simulated by software through writing to the Interrupt Set Shadow register (MCANSS_ISS). The MCANSS_ISS register is a shadow register mapped to the same address as the MCANSS_IRS register.

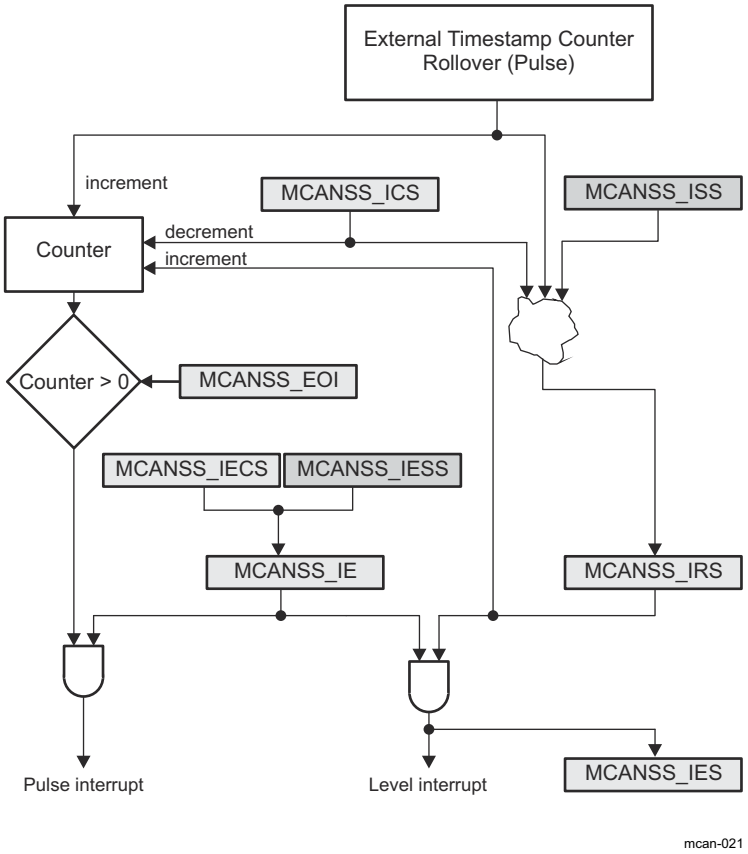


Figure 34-12. External Timestamp Counter Interrupt

34.5.11 Timeout Counter

The MCAN module has an integrated 16-bit timeout counter. The timeout counter is used to signal timeout conditions for the Rx FIFO 0, Rx FIFO 1, and Tx Event FIFO Message RAM elements. The timeout counter is configured using the MCAN_TOCC register and is enabled using the MCAN_TOCC.ETOC bit. The timeout counter operates as down-counter and uses the same prescaler programmed by the MCAN_TSCC.TCP field as the timestamp counter. The actual counter value can be monitored from the MCAN_TOCV.TOC field. The timeout counter can be started only when MCAN_CCCR.INIT = 0 and stopped when MCAN_CCCR.INIT = 1 (example: when the MCAN enters Bus_Off state). The operation mode is selected by the MCAN_TOCC.TOS field. When continuous mode is selected, the counter starts when MCAN_CCCR.INIT = 0, a write to the MCAN_TOCV register presets the counter to the value configured by the MCAN_TOCC.TOP field and continues down-counting.

In case the timeout counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by the MCAN_TOCC.TOP field. Down-counting is started when the first FIFO element is stored. Writing to the MCAN_TOCV register has no effect. When the counter reaches zero, the interrupt MCAN_IR.TOO flag is set.

In continuous mode, the counter is immediately restarted at the value configured by the MCAN_TOCC.TOP field.

Note

The clock signal for the timeout counter is derived from the CAN core sample point signal. Therefore, the point in time where the timeout counter is decremented can vary due to the synchronization/re-synchronization mechanism of the CAN core. If the baud rate switch feature in CAN FD is used, the timeout counter is clocked differently in arbitration and data field.

34.5.12 Safety

The Message Memory is wrapped in an ECC wrapper providing SECDED parity functionality. The ECC wrapper is controlled by an ECC aggregator.

34.5.12.1 ECC Wrapper

The ECC wrapper provides single error correction (SEC) and double error detection (DED) parity to the message memory content. The ECC wrapper has side band signals for error notification. The ECC Wrapper implements an error injection test mode.

The error correction is done using a lazy write back. When an error is detected, the error is noted in a FIFO queue that waits for an access gap to write the data back and refresh the memory. If a transaction writes new data to the compromised entry before the lazy write back completes, the write back is discarded.

34.5.12.2 ECC Aggregator

This section describes the functional details of the ECC aggregator module.

34.5.12.2.1 ECC Aggregator Overview

The ECC aggregator module supports the following general features:

- Provides a mechanism to control and monitor the ECC RAM in the MCAN module.
- Provides software access to all the ECC related registers.
- Supports software readable status of ECC single/double-bit errors and associated info such as RAM address and data bits that are in error.
- Aggregates level pending status from the ECC RAM into a single interrupt to the Host CPU.

The following feature is not supported:

- Statistics such as tracking the number of single and double-bit errors. If needed, these operations can be handled by software.

34.5.12.2.2 ECC Aggregator Registers

There are three groups of registers in the ECC aggregator module:

- **Global registers:** Aggregator Revision Register (MCANERR_REV), ECC Vector Register (MCANERR_VECTOR), Misc Status Register (MCANERR_STAT), ECC Control Register (MCANERR_CTRL), and ECC Wrapper Revision Register (MCANERR_WRAP_REV).
- **Control and status registers:** ECC Error Control Registers (MCANERR_ERR_CTRL1 and MCANERR_ERR_CTRL2) and ECC Error Status Registers (MCANERR_ERR_STAT1, MCANERR_ERR_STAT2, and MCANERR_ERR_STAT3).
- **Interrupt registers:** interrupt status, interrupt enable set, interrupt enable clear, and EOI (End Of Interrupt) registers that are part of a standard interrupt module. For more information, see the following registers:
  - MCANERR_SEC_EOI
  - MCANERR_SEC_STATUS
  - MCANERR_SEC_ENABLE_SET
  - MCANERR_SEC_ENABLE_CLR
  - MCANERR_DED_EOI
  - MCANERR_DED_STATUS
  - MCANERR_DED_ENABLE_SET
  - MCANERR_DED_ENABLE_CLR

34.5.12.3 Reads to ECC Control and Status Registers

The reads to the ECC control and status registers are triggered by writing a 'read message' to the ECC Vector Register as:

- Software writes value (the ECC RAM ID) to the MCANERR_VECTOR.ECC_VECTOR field to select the ECC RAM for control or status.
- Software writes 1 to the MCANERR_VECTOR.RD_SVBUS bit to trigger a read.
- Software writes read address to the MCANERR_VECTOR.RD_SVBUS_ADDRESS field.
- Software then polls the MCANERR_VECTOR.RD_SVBUS_DONE bit to check if the bit is 1. This bit indicates that the read operation has completed.
- Software reads the data from the ECC control or status register. The following clock cycle (MCAN_ICLK) returns the read data.

34.5.12.4 ECC Interrupts

The ECC aggregator module aggregates the level pending status from the ECC RAM into a single EOI-handshake based interrupt to the Host CPU. Software is expected to follow the sequence described:

- Software enables the interrupts for the ECC RAM by writing to the MCANERR_SEC_ENABLE_SET/ MCANERR_DED_ENABLE_SET register.
- Software writes the ECC RAM ID in the MCANERR_VECTOR.ECC_VECTOR.
- Software writes the MCANERR_VECTOR.RD_SVBUS bit to trigger the read.
- Software writes the MCANERR_ERR_STAT1 register address to the MCANERR_VECTOR.RD_SVBUS_ADDRESS field. Software needs to load the 'read message' in the rMCANERR_VECTOR register again, if the software needs to read the MCANERR_ERR_STAT2 register.
- Software polls the MCANERR_VECTOR.RD_SVBUS_DONE bit. When this bit is set, a read of the MCANERR_ERR_STAT1/MCANERR_ERR_STAT2 register is performed.
- After the interrupt has been serviced, software clears the interrupt status by writing to the MCANERR_ERR_STAT1.CLR_ECC_SEC or MCANERR_ERR_STAT1.CLR_ECC_DED bit depending on the type of the ECC error.
- Software polls the MCANERR_ERR_STAT1 register to verify that the status bit has been cleared.
- Software writes to the MCANERR_SEC_EOI/MCANERR_DED_EOI register to clear the interrupt.
- After clearing the ECC interrupt source, the application software must also write 1 to the MCANERR_SEC_EOI.EOI_WR /MCANERR_DED_EOI.EOI_WR bits.

34.5.13 Rx Handling

The Rx Handler controls the following operations:

- Acceptance filtering
- The transfer of received messages to the Rx buffers or to one of the two Rx FIFOs (Rx FIFO 0 or Rx FIFO 1)
- Rx FIFO Put and Get Index operations

34.5.13.1 Acceptance Filtering

The MCAN module employs two sets of acceptance filters - one set for standard and one set for extended identifiers. These filters can be assigned to an Rx Buffer or to one of the two Rx FIFOs.

The main features of the filter elements are:

- Each filter element can be configured as:
  - Range filter (from - to)
  - Filter for specific IDs (for one or two dedicated IDs)
  - Classic bit mask filter
- Each filter element can be enabled/disabled individually
- Each filter element can be configured for acceptance or rejection filtering
- Filters are checked sequentially and execution (acceptance filtering procedure) stops at the first matching filter element or when the end of the filter list is reached

Related configuration registers are:

- Global Filter Configuration (MCAN_GFC) register
- Standard ID Filter Configuration (MCAN_SIDFC) register
- Extended ID Filter Configuration (MCAN_XIDFC) register
- Extended ID AND Mask (MCAN_XIDAM) register

Depending on the configuration of the filter element (see SFEC/EFEC in Section 34.5.16) if filter matches, one of the following actions is performed:

- Received frame is stored in FIFO 0 or FIFO 1
- Received frame is stored in Rx Buffer
- Received frame is stored in Rx Buffer and generation of pulse at filter event pin is performed. This is high level single MCAN_ICLK pulse.
- Received frame is rejected
- Set High Priority Message interrupt flag MCAN_IR.HPM
- Set High Priority Message interrupt flag MCAN_IR.HPM and store received frame in FIFO 0 or FIFO 1

Acceptance filtering starts when complete Message ID is received. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If a filter element matches - the Rx Handler starts writing the received message data in portions of 32-bit to the matching Rx Buffer or Rx FIFO. If an error condition occurs (for example: CRC error), this message is rejected with the following impact on the affected Rx Buffer or Rx FIFO:

- Rx Buffer: New Data flag (MCAN_NDAT1/MCAN_NDAT2) of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data (for error type, see MCAN_PSR.LEC and MCAN_PSR.DLEC fields,respectively).
- Rx FIFO: Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data (for error type, see MCAN_PSR.LEC and MCAN_PSR.DLEC fields, respectively). If matching Rx FIFO is configured to operate in overwrite mode, the boundary conditions described in Section 34.5.13.2.2 must be considered.

Note

When an accepted message is written to one of the two Rx FIFOs, or into an Rx Buffer, the unmodified received identifier is stored independent of the filters used. The result of the acceptance filter process is strongly depending on the sequence of configured filter elements.

34.5.13.1.1 Range Filter

Each filter element can be configured to operate as Range Filter (Standard Filter Type SFT = 00/Extended Filter Type EFT = 00). The filter matches for all received message frames with IDs in the range from SFID1 to SFID2 (SFID2 ≥ SFID1) respectively in the range from EFID1 to EFID2 (EFID2 ≥ EFID1). For more information see Section 34.5.16.5 and Section 34.5.16.6.

There are two options for range filtering of extended frames:

- Extended Filter Type EFT = 00: The Extended ID AND Mask (MCAN_XIDAM) is used for Range Filtering. The Message ID of received frames is ANDed with the Extended ID AND Mask (MCAN_XIDAM) before the range filter is applied.
- Extended Filter Type EFT = 11: The Extended ID AND Mask (MCAN_XIDAM) is not used for Range Filtering.

34.5.13.1.2 Filter for Specific IDs

Each filter element can be configured to filter one or two dedicated Message IDs (Standard Filter Type SFT =01/ Extended Filter Type EFT =01). To filter only one specific Message ID, the filter element has to be configured with SFID1 = SFID2 respectively EFID1 = EFID2. For more information, see Section 34.5.16.5 and Section 34.5.16.6.

34.5.13.1.3 Classic Bit Mask Filter

Classic bit mask filtering can filter groups of Message IDs (Standard Filter Type SFT =10/Extended Filter Type EFT =10). This is done by masking single bits of a received Message ID. In this case SFID1/EFID1 element is used as Message ID filter, while the SFID2/EFID2 element is used as filter mask.

A 0 bit at the filter mask (SFID2/EFID2) masks out the corresponding bit position of the configured Message ID filter (SFID1/EFID1) and the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are 1 are relevant for acceptance filtering.

There are two interesting cases:

- All mask bits are 1: a match occurs only when the received Message ID and the configured Message ID filter are identical.
- All mask bits are 0: all Message IDs match.

34.5.13.1.4 Standard Message ID Filtering

Figure 34-13 shows the standard Message ID (11-bit ID) filtering flow. Section 34.5.16.5 describes the standard Message ID filter element.

The Remote Transmission Request (RTR) and Extended Identifier (XTD) bits of the received frames are compared against the list of configured filter elements. This is controlled by the following registers:

- Global Filter Configuration (MCAN_GFC) register
- Standard ID Filter Configuration (MCAN_SIDFC) register

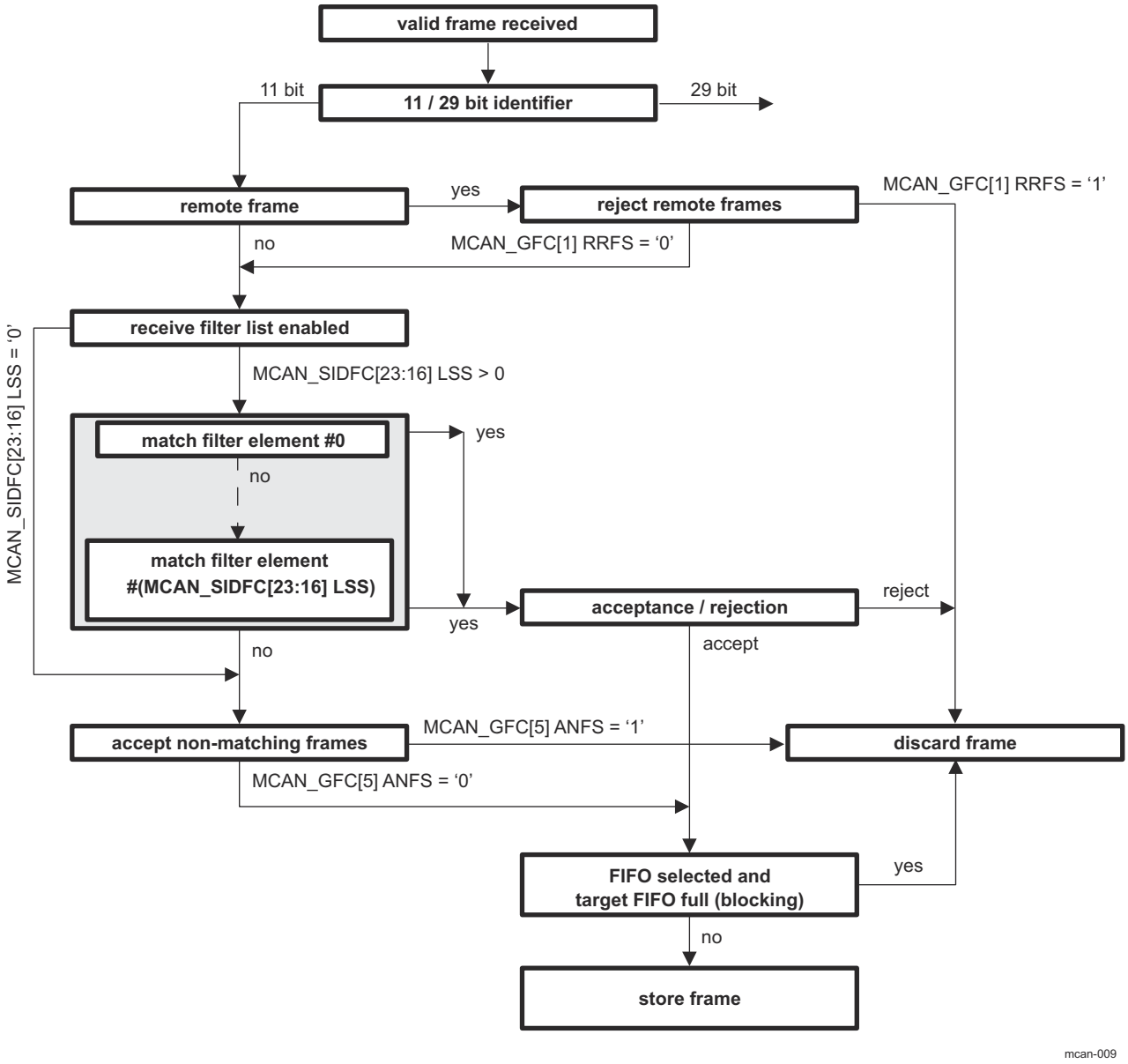


Figure 34-13. Standard Message ID Filter Path

34.5.13.1.5 Extended Message ID Filtering

Figure 34-14 shows the extended Message ID (29-bit ID) filtering flow. Section 34.5.16.6 describes the extended Message ID filter element.

The Remote Transmission Request (RTR) and Extended Identifier (XTD) bits of the received frames are compared against the list of configured filter elements. This is controlled by the following registers:

- Global Filter Configuration (MCAN_GFC) register
- Extended ID Filter Configuration (MCAN_XIDFC) register

Note that before the filter list is executed, the received identifier is ANDed with the Extended ID AND Mask (MCAN_XIDAM).

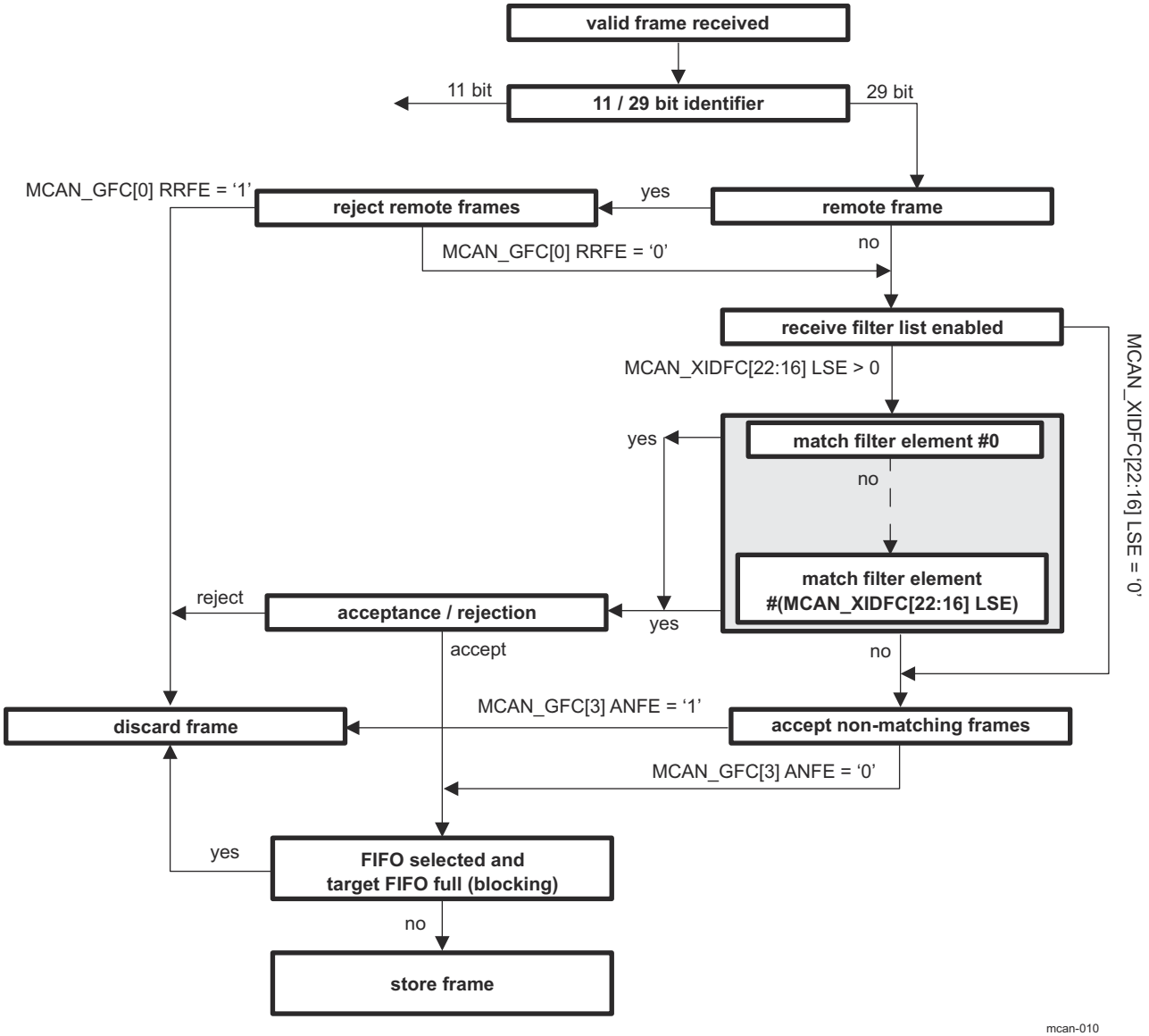


Figure 34-14. Extended Message ID Filter Path

34.5.13.2 Rx FIFOs

The configuration of the Rx FIFOs (Rx FIFO 0 and Rx FIFO 1) can be done by way of the MCAN_RXF0C and MCAN_RXF1C registers. Each Rx FIFO can be configured to store up to 64 received messages.

After acceptance filtering the received messages that passed are transferred to the Rx FIFO. The filter mechanisms available for the Rx FIFO 0 and Rx FIFO 1 are described in Section 34.5.13.1. The Rx FIFO element is described in Section 34.5.16.2.

The Rx FIFO watermark can be used to prevent an Rx FIFO overflow. If the Rx FIFO fill level reaches the Rx FIFO watermark configured by the MCAN_RXFnC[30:24].FnWM bit (where: n = 0 or 1), an interrupt flag MCAN_IR.RF0W/MCAN_IR.RF1W is set.

When the Rx FIFO Put Index reaches the Rx FIFO Get Index (MCAN_RXFnS[21:16].FnPI = MCAN_RXFnS[13:8].FnGI), an Rx FIFO Full condition is signaled by the MCAN_RXFnS[24].FnF status bit and interrupt flag MCAN_IR.RF0F/MCAN_IR.RF1F is set. Figure 34-15 shows Rx FIFO Status. The FIFOs fill level is presented in the MCAN_RXFnS[6:0].FnFL field (the number of elements stored in Rx FIFO).

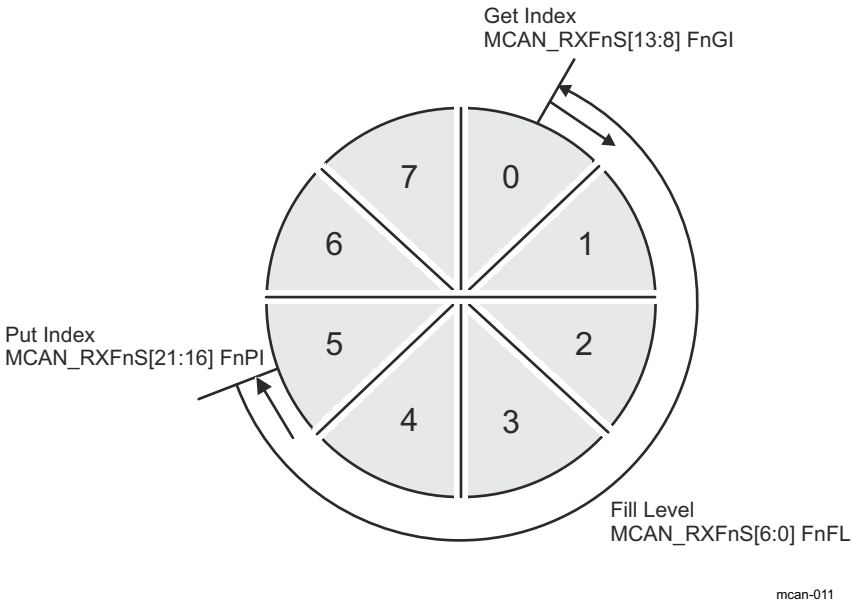


Figure 34-15. Rx FIFO Status

Rx FIFOs start address in the Message RAM (MCAN_RXFnC[15:2].FnSA field) has to be configured when reading from an Rx FIFO (Rx FIFO Get Index - MCAN_RXFnS[13:8].FnGI). Table 34-6 presents Rx Buffer/Rx FIFO Element Size for different Rx Buffer/Rx FIFO Data Field Size which is configured by way of the MCAN_RXESC register.

Table 34-6. Rx Buffer/Rx FIFO Element Size

MCAN_RXESC Register RBDS/F0DS/F1DS Bits	Data Field [bytes]	FIFO Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

34.5.13.2.1 Rx FIFO Blocking Mode

The Rx FIFO blocking mode is the default operation mode for the Rx FIFOs and is configured by MCAN_RXFnC[31].FnOM = 0.

If an Rx FIFO full condition is reached (MCAN_RXFnS[21:16].FnPI = MCAN_RXFnS[13:8].FnGI), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signaled by the MCAN_RXFnS[24].FnF = 1 and interrupt flag MCAN_IR.RF0F/MCAN_IR.RF1F is set.

In case a message is received while the corresponding Rx FIFO is full, this message is rejected and the message lost condition is signaled by MCAN_RXFnS[25].RFnL = 1 and interrupt flag MCAN_IR.RF0L/ MCAN_IR.RF1L is set.

34.5.13.2.2 Rx FIFO Overwrite Mode

The Rx FIFO overwrite mode is configured by MCAN_RXFnC[31].FnOM = 1. When an Rx FIFO full condition is reached (MCAN_RXFnS[21:16].FnPI = MCAN_RXFnS[13:8].FnGI) signaled by MCAN_RXFnS[24].FnF = 1, the next accepted message for the FIFO overwrites the oldest FIFO message. Put index/Get index are both incremented by one.

In overwrite mode if an Rx FIFO full condition is signaled, reading of the Rx FIFO elements starts at least at get index + 1. The reason for this is a received message is written to the Message RAM (Put index) while the Host CPU is reading from the Message RAM (Get index). In this case, inconsistent data can be read from the respective Rx FIFO element. The problem is solved by adding an offset to the Get index when reading from the Rx FIFO. The offset depends on how fast the Host CPU accesses the Rx FIFO. Figure 34-16 shows an offset of two with respect to the Get index when reading the Rx FIFO. In this case, the two messages stored in element 1 and 2 are lost.

After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index MCAN_RXFnA[5:0].FnAI. This increments the get index to that element number. In case the Put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (MCAN_RXFnS[24].FnF = 0).

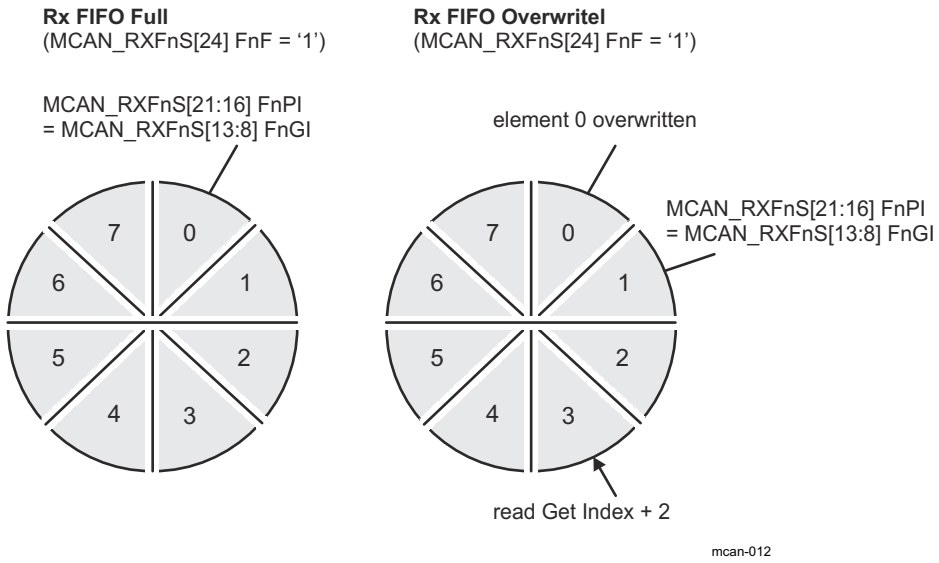


Figure 34-16. Rx FIFO Overflow Handling

34.5.13.3 Dedicated Rx Buffers

The MCAN supports up to 64 dedicated Rx buffers. The start address of the Rx buffers section in the Message RAM is configured by way of the MCAN_RXBC.RBSA field. To store in an Rx Buffer a Standard or Extended Message ID Filter Element with SFEC/EFEC = 111 and SFID2/EFID2[10:9] = 00 has to be configured (see Section 34.5.16.5 and Section 34.5.16.6).

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element (the format is the same as for an Rx FIFO element). In addition, the flag MCAN_IR.DRX (message stored in Dedicated Rx Buffer) is set.

Table 34-7 shows an example filter configuration for Rx buffers.

Table 34-7. Example Filter Configuration for Rx Buffers

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID message 1	00	00 0000
1	ID message 2	00	00 0001
2	ID message 3	00	00 0010

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in register MCAN_NDAT1/MCAN_NDAT1 is set. As long as the New Data flag is set, the respective Rx Buffer is locked against updates from received matching frames. The New Data flags must be reset by the Host CPU by writing a 1 to the respective bit position.

While an Rx buffer New Data flag is set, a Message ID Filter Element referencing this specific Rx Buffer does not match, causing the acceptance filtering to continue. Following Message ID Filter Elements can cause the received message to be stored into another Rx Buffer, into an Rx FIFO, or the message can be rejected, depending on filter configuration.

34.5.13.3.1 Rx Buffer Handling

Rx Buffer Handling include the following steps:

- Reset interrupt flag MCAN_IR.DRX
- Read New Data registers
- Read messages from Message RAM
- Reset New Data flags of processed messages

34.5.14 Tx Handling

The Tx handler is used to handle the Tx requests. The Tx handler controls the transfer of transmit messages from the dedicated Tx buffers, the Tx FIFO, and the Tx Queue to the CAN Core, the Tx Event FIFO, and the Put and Get Index operations. The MCAN module supports up to 32 Tx buffers. These Tx buffers can be configured as dedicated Tx buffers, Tx FIFO, or Tx Queue and as combination of dedicated Tx buffers/Tx FIFO or dedicated Tx buffers/Tx Queue. For each Tx Buffer element Classical CAN or CAN FD transmission mode can be configured. Section 34.5.16.3 describes the Tx Buffer Element. Table 34-8 shows the possible configurations for message transmission.

Table 34-8. Possible Configurations for Message Transmission

MCAN_CCCR Register		Tx Buffer Element		Frame Transmission
BRSE	FDOE	FDF	BRS	
ignored	0	ignored	ignored	Classic CAN
0	1	0	ignored	Classic CAN
0	1	1	ignored	CAN FD without bit rate switching
1	1	0	ignored	Classic CAN
1	1	1	0	CAN FD without bit rate switching
1	1	1	1	CAN FD with bit rate switching

When the Tx Buffer Request Pending (MCAN_TXBRP) register is updated, or when a transmission has been started the Tx Handler starts scanning to check for the highest priority pending Tx request. The Tx Buffer with the lowest Message ID has highest priority.

Note

AUTOSAR requires at least three Tx Queue buffers and support of transmit cancellation.

34.5.14.1 Transmit Pause

The transmit pause feature is intended for use in CAN networks where the CAN Message IDs are specific and cannot easily be changed. These Message IDs can have a higher priority than other defined Message IDs, while in a specific application the relative priority can be inverse. This allows for a case where one ECU sends a burst of CAN messages that cause another ECU CAN messages to be delayed (paused).

The transmit pause feature is enabled by the MCAN_CCCR.TXP bit. By default this bit is disabled (MCAN_CCCR.TXP = 0). Each time after successfully transmitted message, a pause for two CAN bit times occurs before the start of the next transmission. This allows the other CAN nodes in the network to transmit messages even if the Message IDs have lower priority.

34.5.14.2 Dedicated Tx Buffers

Dedicated Tx buffers are intended for message transmission under complete control of the Host CPU.

There are two options:

- Each dedicated Tx Buffer is configured with a specific Message ID.
- Two or more dedicated Tx buffers are configured with the same Message ID. In this case the Tx Buffer with the lowest buffer number is transmitted first.

After the data section has been updated, a transmission is requested by an Add Request. This is done using the MCAN_TXBAR[x]ARn bit (where x = 0 to 31). The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to the Message ID.

Table 34-9 shows Tx Buffer/Tx FIFO/Tx Queue Element Size. A Dedicated Tx Buffer allocates element size 32-bit words in the Message RAM. The start address of a dedicated Tx Buffer in the Message RAM is calculated by adding transmit buffer index from 0 to 31 (MCAN_TXFQS.TFQP) × Element size to the Tx Buffer start address MCAN_TXBC.TBSA field.

Table 34-9. Tx Buffer, Tx FIFO, Tx Queue Element Size

MCAN_TXESC.TBDS	Data Field (bytes)	Element Size (RAM Words)
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

34.5.14.3 Tx FIFO

Tx FIFO mode is configured by setting bit MCAN_TXBC.TFQM = 0. The stored in the Tx FIFO messages are transmitted starting with the message referenced by the Get Index MCAN_TXFQS.TFGI field. After each transmission the Get Index is incremented until the Tx FIFO is empty. The Tx FIFO Free Level MCAN_TXFQS.TFFL field indicates the number of the available free Tx FIFO elements. The Tx FIFO allows transmission of messages with the same Message ID from different Tx buffers in the order these messages have been written to the Tx FIFO.

New transmit messages must be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index MCAN_TXFQS.TFQP field. After each Add Request (MCAN_TXBAR[x] ARn = 1), the Put Index is incremented to the next free Tx FIFO element. When the Put Index reaches the Get Index (MCAN_TXFQS.TFQP = MCAN_TXFQS.TFGI), Tx FIFO Full condition is signaled by bit MCAN_TXFQS.TFQF = 1. In this case, no further messages must be written to the Tx FIFO until the next message is transmitted and the Get Index is incremented.

The number of requested Tx buffers must not exceed the number of free Tx buffers, as indicated by the Tx FIFO Free Level MCAN_TXFQS.TFFL field.

In case a transmission request for the Tx Buffer referenced by the Get Index is canceled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level MCAN_TXFQS.TFFL field is recalculated. In case transmission cancellation is applied to any other Tx Buffer, the Get Index and the FIFO Free Level remain unchanged.

A Tx FIFO element allocates element size 32-bit words in the Message RAM (see Table 34-9). The start address of the next available (free) Tx FIFO Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS.TFQP (from 0 to 31) × Element Size to the Tx Buffer Start Address MCAN_TXBC.TBSA field.

34.5.14.4 Tx Queue

Tx Queue mode is configured by setting bit MCAN_TXBC.TFQM = 1. The stored in the Tx Queue messages are transmitted starting with the highest priority message (lowest Message ID). In case two or more Queue buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New transmit messages must be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index MCAN_TXFQS.TFQP field. Each Add Request cyclically increments the Put Index to the next free Tx Buffer. In case of Tx Queue Full condition (MCAN_TXFQS.TFQF = 1), the Put Index is not valid and no further message must be written to the Tx Queue until at least one of the requested messages is sent out or a pending transmission request is canceled.

The application can use the MCAN_TXBRP register instead of the Put Index and can place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates element size 32-bit words in the Message RAM (see Table 34-9). The start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS.TFQP (from 0 to 31) × Element Size to the Tx Buffer Start Address MCAN_TXBC.TBSA field.

34.5.14.5 Mixed Dedicated Tx Buffers/Tx FIFO

For this combination the Tx buffers section in the Message RAM is separated in two parts:

- Dedicated Tx buffers: the number of Dedicated Tx buffers is configured by the MCAN_TXBC.NDTB field
- Tx FIFO: the number of Tx buffers assigned to the Tx FIFO is configured by the MCAN_TXBC.TFQS field

If the MCAN_TXBC.TFQS field is empty (zero) - only Dedicated Tx buffers are used.

Tx prioritization:

- Scan Dedicated Tx buffers and oldest pending Tx FIFO Buffer (referenced by the MCAN_TXFQS.TFGI field)
- Buffer with lowest Message ID gets highest priority and is transmitted next

Figure 34-17 shows Mixed Dedicated Tx buffers/Tx FIFO example.

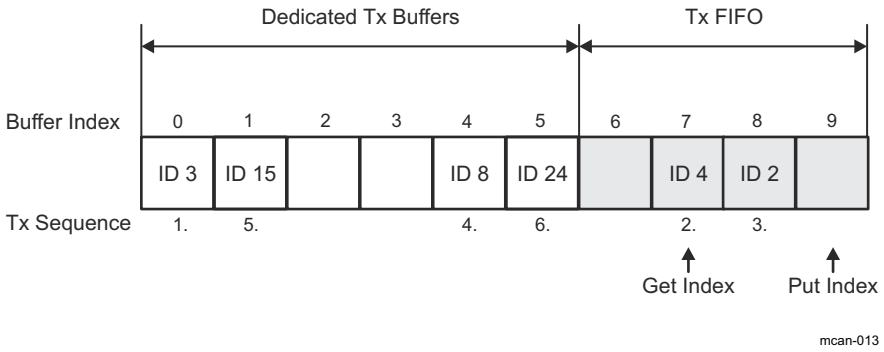


Figure 34-17. Mixed Dedicated Tx Buffers /Tx FIFO (example)

34.5.14.6 Mixed Dedicated Tx Buffers/Tx Queue

For this combination the Tx buffers section in the Message RAM is separated in two parts:

- Dedicated Tx buffers: the number of Dedicated Tx buffers is configured by the MCAN_TXBC.NDTB field
- Tx Queue: the number of Tx buffers assigned to the Tx Queue is configured by the MCAN_TXBC.TFQS field

If MCAN_TXBC.TFQS field is empty (zero) - only Dedicated Tx buffers are used.

Tx prioritization:

- Scan all Tx buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next

Figure 34-18 shows Mixed Dedicated Tx buffers/Tx Queue example.

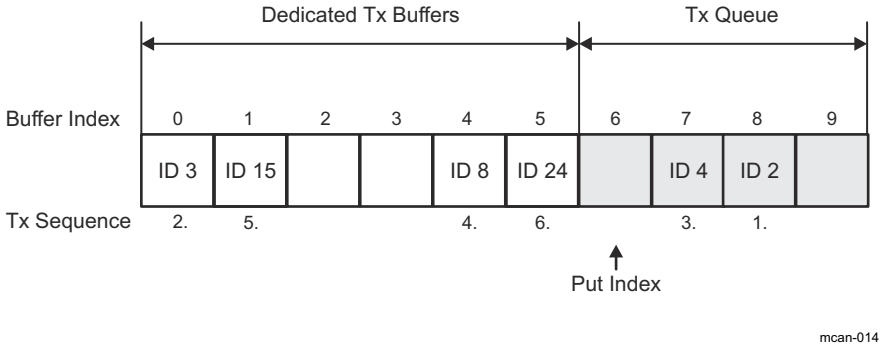


Figure 34-18. Mixed Dedicated Tx Buffers /Tx Queue (example)

34.5.14.7 Transmit Cancellation

This feature is especially intended for gateway and AUTOSAR based applications. The Host CPU can cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer by setting bit MCAN_TXBCR[n].CRn = 1 (where n = 0 to 31). The corresponding bit position n is equivalent to the number of the Tx buffer.

Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signaled by setting the corresponding bit of the MCAN_TXBCF register (MCAN_TXBCF[n].CFn = 1).

If transmission from a Tx Buffer is already ongoing and a transmit cancellation is requested, the corresponding MCAN_TXBRP[n].TRPn bit remains set as long as the transmission is in progress. If the transmission was

successful, the corresponding MCAN_TXBTO[n].TOn and MCAN_TXBCF[n].CFn bits are set. If the transmission was not successful, only the corresponding bit MCAN_TXBCF[n].CFn = 1.

Note

If a pending transmission is canceled immediately before this transmission has started, a short time window occurs where no transmission is started even if another message is also pending in this node. This can enable another node to transmit a message that can have a lower priority than the second message in this node.

34.5.14.8 Tx Event Handling

To support Tx Event Handling, the Message RAM has implemented a Tx Event FIFO section. Up to 32 Tx Event FIFO elements can be configured. Section 34.5.16.4 describes the Tx Event FIFO element. After message transmission on the CAN bus, Message ID and Timestamp are stored in a Tx Event FIFO element. To link a Tx Event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

A Tx Event FIFO full condition is signaled by the MCAN_IR.TEFF bit. In this case no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index has been incremented (MCAN_TXEFS.EFGI). In case a Tx Event occurs while the Tx Event FIFO is full, this event is rejected and interrupt flag MCAN_IR.TEFL bit is set.

The Tx Event FIFO watermark can be configured to avoid a Tx Event FIFO overflow. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by the MCAN_TXEFC.EFWM field, interrupt flag MCAN_IR.TEFW is set. When reading from the Tx Event FIFO, two times the Tx Event FIFO Get Index MCAN_TXEFS.EFGI field has to be added to the Tx Event FIFO start address MCAN_TXEFC.EFSA field.

34.5.15 FIFO Acknowledge Handling

The Get Indices of the two Rx FIFOs (Rx FIFO 0 or Rx FIFO 1) and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index (see MCAN_RXF0A, MCAN_RXF1A, and MCAN_TXEFA). Writing to the FIFO Acknowledge Index sets the FIFO Get Index to the FIFO Acknowledge Index plus one and, thereby, updates the FIFO Fill Level.

There are two use cases:

- A single element has been read from the FIFO: the Get Index value is written to the FIFO Acknowledge Index.
- A sequence of elements has been read from the FIFO: the Get Index value (Index of the last element read) is written to the FIFO Acknowledge Index at the end of that read sequence.

The Host CPU has free access to the Message RAM. Special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This can be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case, the FIFO Acknowledge Index must not be written because this sets the Get Index to a wrong position and also changes the FIFO's Fill Level. In this case, some of the older FIFO elements can be lost.

Note

The application has to make sure that a valid value is written to the FIFO Acknowledge Index. The MCAN module does not check for erroneous values.

34.5.16 Message RAM

The MCAN module has a Message RAM. The main purpose of the Message RAM is to store:

- Received Messages
- Transmit Messages
- Tx Event Elements
- Message ID Filter Elements

34.5.16.1 Message RAM Configuration

The MCAN module is configured to allocate up to 1024 words in the Message RAM. The Message RAM has a width of 32 bits.

The address ranges of the Message RAMs are from 0x6002 0000 to 0x6002 0FFF, 0x6002 8000 to 0x6002 8FFF, 0x6003 0000 to 0x6003 0FFF, 0x6003 8000 to 0x6003 8FFF, 0x6004 0000 to 0x6004 0FFF, and from 0x6004 8000 to 0x6004 8FFF.

The Message RAM is capable to include each of the sections listed in Figure 34-19. It is not necessary to configure each of the sections (a section in the Message RAM can be 0) and there is no restriction with respect to the sequence of the sections. For parity checking or ECC, a respective number of bits has to be added to each word. When the MCAN module addresses the Message RAM, the MCAN addresses 32-bit words. The start addresses are configurable and are 32-bit word addresses.

The element size can be configured for:

- Rx FIFO 0, by way of the MCAN_RXESC.F0DS field
- Rx FIFO, 1 by way of the MCAN_RXESC.F1DS field
- Rx buffers, by way of the MCAN_RXESC.RBDS field
- Tx buffers, by way of the MCAN_TXESC.TBDS field

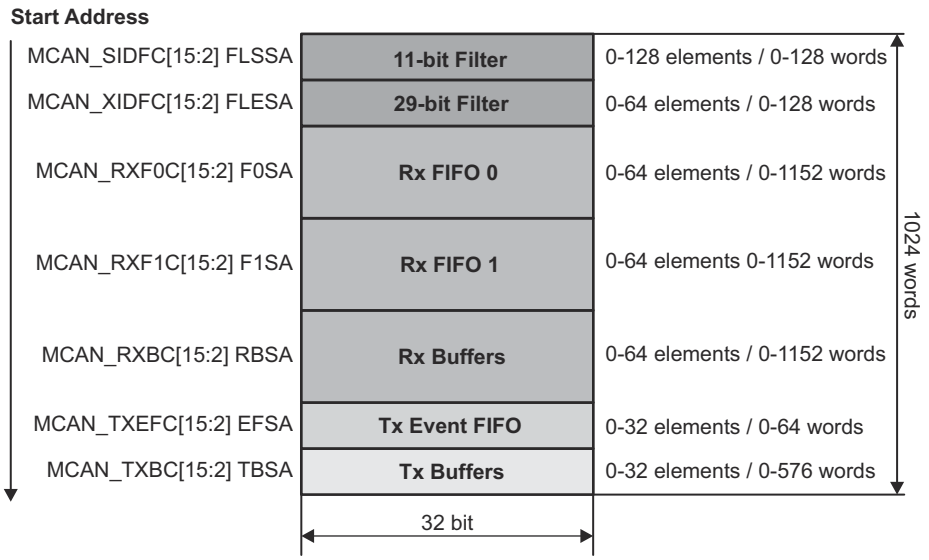


Figure 34-19. Message RAM Configuration

The Host CPU configures the following information in the Message RAM:

- Start addresses of the memory sections
- Number of elements in each section
- The size of the elements in some sections

Note

The MCAN module does not check for errors in the Message RAM configuration. The configuration of the start addresses of the different sections and the number of elements of each section has to be done carefully. This prevents falsification or loss of data.

34.5.16.2 Rx Buffer and FIFO Element

Up to 64 Rx buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field by way of the MCAN_RXESC register.

Figure 34-20 shows the Rx Buffer/Rx FIFO element structure. Table 34-10 shows the Rx Buffer/Rx FIFO element field descriptions.

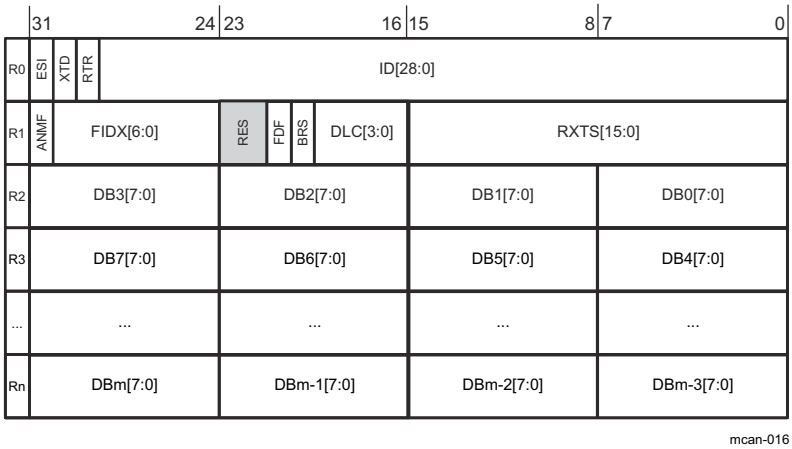


Figure 34-20. Rx Buffer/Rx FIFO Element Structure

Table 34-10. Rx Buffer/Rx FIFO Element Field Descriptions

Word	Bits	Field Name	Description
R0	31	ESI	Error State Indicator • 0x0: Transmitting node is error active • 0x1: Transmitting node is error passive
	30	XTD	Extended Identifier Signals to the Host CPU whether the received frame has a standard or extended identifier. • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request Signals to the Host CPU whether the received frame is a data frame or a remote frame. • 0x0: Received frame is a data frame • 0x1: Received frame is a remote frame **Note:** There are no remote frames in CAN FD format. In case a CAN FD frame was received (FDF = 1), RTR bit reflects the state of the reserved r1 bit (RES[23]). In CAN FD frames (FDF = 1), the dominant RRS (Remote Request Substitution) bit replaces the RTR (Remote Transmission Request) bit.
	28:0	ID[28:0]	Identifier Standard or extended identifier depending on XTD bit. A standard identifier is stored into ID[28:18].

Table 34-10. Rx Buffer/Rx FIFO Element Field Descriptions (continued)

Word	Bits	Field Name	Description
R1	31	ANMF	Accepted Non-matching Frame Acceptance of non-matching frames can be enabled using the MCAN_GFC.ANFS and MCAN_GFC.ANFE fields. • 0x0: Received frame matching filter index FIDX field • 0x1: Received frame did not match any Rx filter element
	30:24	FIDX[6:0]	Filter Index 0x0-0x7F (0-127): Index of matching Rx acceptance filter element (invalid if ANMF = 1). Range is 0 to MCAN_SIDFC.LSS - 1 respectively MCAN_XIDFC.LSE - 1.
	23:22	RES	Reserved
	21	FDF	FD Format • 0x0: Standard frame format • 0x1: CAN FD frame format (new DLC-coding and CRC)
	20	BRS	Bit Rate Switch • 0x0: Frame received without bit rate switching • 0x1: Frame received with bit rate switching
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: received frame has 0-8 data bytes • 0x9-0xF (9-15): CAN: received frame has 8 data bytes • 0x9-0xF (9-15): CAN FD: received frame has 12/16/20/24/32/48/64 data bytes
	15:0	RXTS[15:0]	Rx Timestamp Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC.TCP.
R2	31:24	DB3[7:0]	Data Byte 3
	23:16	DB2[7:0]	Data Byte 2
	15:8	DB1[7:0]	Data Byte 1
	7:0	DB0[7:0]	Data Byte 0
R3	31:24	DB7[7:0]	Data Byte 7
	23:16	DB6[7:0]	Data Byte 6
	15:8	DB5[7:0]	Data Byte 5
	7:0	DB4[7:0]	Data Byte 4
...	...	...	...
Rn	31:24	DBm[7:0]	Data Byte m
	23:16	DBm-1[7:0]	Data Byte m-1
	15:8	DBm-2[7:0]	Data Byte m-2
	7:0	DBm-3[7:0]	Data Byte m-3

Note

Depending on the configuration of the element size (MCAN_RXESC), between two and sixteen 32-bit words (Rn = 3-17) are used for storage of a CAN message's data field.

34.5.16.3 Tx Buffer Element

The Tx buffers section can be configured to hold dedicated Tx buffers as well as a Tx FIFO/Tx Queue. In case that the Tx buffers section is shared by dedicated Tx buffers and a Tx FIFO/Tx Queue, the dedicated Tx buffers start at the beginning of the Tx buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler makes difference between dedicated Tx buffers and Tx FIFO/Tx Queue by way of the MCAN_TXBC.TFQS and MCAN_TXBC.NDTB fields. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field by way of the MCAN_TXESC register.

Figure 34-21 shows the Tx Buffer element structure. Table 34-11 shows the Tx Buffer element field descriptions.

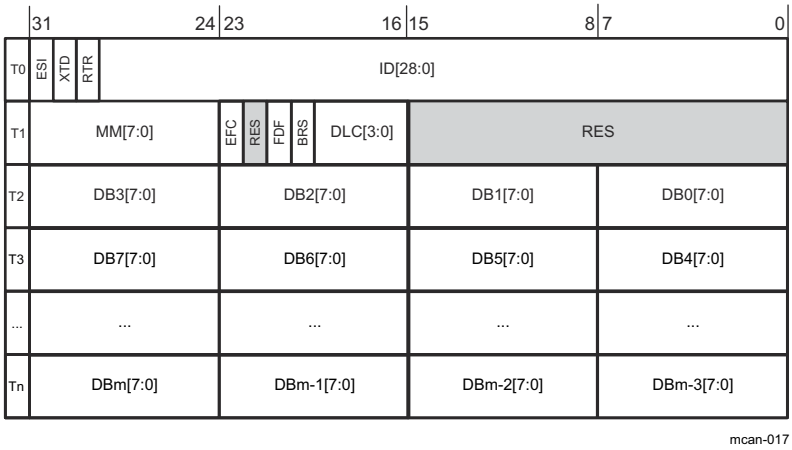


Figure 34-21. Tx Buffer Element Structure

Table 34-11. Tx Buffer Element Field Descriptions

Word	Bits	Field Name	Description
T0	31	ESI	Error State Indicator • 0x0: ESI bit in CAN FD format depends only on error passive flag • 0x1: ESI bit in CAN FD format transmitted recessive **Note:** The ESI bit of the transmit buffer is ORed with the error passive flag to decide the value of the ESI bit in the transmitted CAN FD frame. As required by the CAN FD protocol specification, an error active node can optionally transmit the ESI bit recessive, but an error passive node always transmits the ESI bit recessive.
	30	XTD	Extended Identifier • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request • 0x0: Transmit data frame • 0x1: Transmit remote frame **Note:** When RTR = 1, the MCAN module transmits a remote frame according to ISO11898-1:2015, even if the MCAN_CCCR.FDOE bit enables the transmission in CAN FD format.
	28:0	ID[28:0]	Identifier Standard or extended identifier depending on XTD bit. A standard identifier has to be written to ID[28:18].

Table 34-11. Tx Buffer Element Field Descriptions (continued)

Word	Bits	Field Name	Description
T1	31:24	MM[7:0]	Message Marker Written by Host CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status (see also MM[7:0] field in Table 34-12).
	23	EFC	Event FIFO Control • 0x0: Don't store Tx events • 0x1: Store Tx events
	22	RES	Reserved
	21	FDF	FD Format • 0x0: Frame transmitted in Classic CAN format • 0x1: Frame transmitted in CAN FD format
	20	BRS	Bit Rate Switch • 0x0: CAN FD frames transmitted without bit rate switching • 0x1: CAN FD frames transmitted with bit rate switching **Note:** ESI, FDF, and BRS bits are only evaluated when CAN FD operation is enabled using the MCAN_CCCR.FDOE bit. BRS bit is only evaluated when MCAN_CCCR.BRSE = 1.
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: transmit frame has 0-8 data bytes • 0x9-0xF (9-15): CAN: transmit frame has 8 data bytes • 0x9-0xF (9-15): CAN FD: transmit frame has 12/16/20/24/32/48/64 data bytes
	15:0	RES	Reserved
T2	31:24	DB3[7:0]	Data Byte 3
	23:16	DB2[7:0]	Data Byte 2
	15:8	DB1[7:0]	Data Byte 1
	7:0	DB0[7:0]	Data Byte 0
T3	31:24	DB7[7:0]	Data Byte 7
	23:16	DB6[7:0]	Data Byte 6
	15:8	DB5[7:0]	Data Byte 5
	7:0	DB4[7:0]	Data Byte 4
...	...	...	...
Tn	31:24	DBm[7:0]	Data Byte m
	23:16	DBm-1[7:0]	Data Byte m-1
	15:8	DBm-2[7:0]	Data Byte m-2
	7:0	DBm-3[7:0]	Data Byte m-3

Note

Depending on the configuration of the element size (MCAN_TXESC), between two and sixteen 32-bit words (Tn = 3-17) are used for storage of a CAN message's data field.

34.5.16.4 Tx Event FIFO Element

Each element stores information about transmitted messages. By reading the Tx Event FIFO the Host CPU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from the MCAN_TXEFS register.

Figure 34-22 shows the Tx Event FIFO element structure. Table 34-12 shows the Tx Event FIFO element field descriptions.

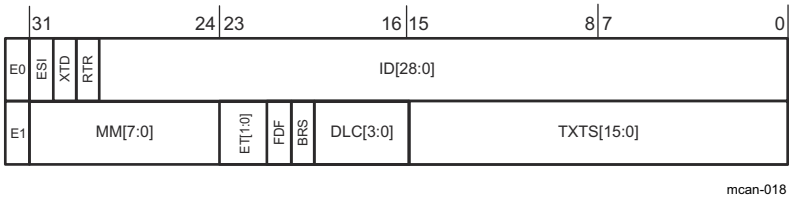


Figure 34-22. Tx Event FIFO Element Structure

Table 34-12. Tx Event FIFO Element Field Descriptions

Word	Bits	Field Name	Description
E0	31	ESI	Error State Indicator • 0x0: Transmitting node is error active • 0x1: Transmitting node is error passive
	30	XTD	Extended Identifier • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request • 0x0: Data frame transmitted • 0x1: Remote frame transmitted
	28:0	ID[28:0]	Identifier Standard or extended identifier depending on XTD bit. A standard identifier has to be written to ID[28:18].

Table 34-12. Tx Event FIFO Element Field Descriptions (continued)

Word	Bits	Field Name	Description
E1	31:24	MM[7:0]	Message Marker Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status (see also MM[7:0] field in Table 34-11).
	23:22	ET[1:0]	Event Type • 0x0: Reserved • 0x1: Tx event • 0x2: Transmission in spite of cancellation (always set for transmissions in DAR mode) • 0x3: Reserved
	21	FDF	FD Format • 0x0: Standard frame format • 0x1: CAN FD frame format (new DLC-coding and CRC)
	20	BRS	Bit Rate Switch • 0x0: Frame transmitted without bit rate switching • 0x1: Frame transmitted with bit rate switching
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: frame with 0-8 data bytes transmitted • 0x9-0xF (9-15): CAN: frame with 8 data bytes transmitted • 0x9-0xF (9-15): CAN FD: frame with 12/16/20/24/32/48/64 data bytes transmitted
	15:0	TXTS[15:0]	Tx Timestamp Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC.TCP filed.

34.5.16.5 Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, the element address is the Filter List Standard Start Address MCAN_SIDFC.FLSSA field plus the index of the filter element (0-127).

Figure 34-23 shows the Standard Message ID Filter element structure. Table 34-13 shows the Standard Message ID Filter element field descriptions.

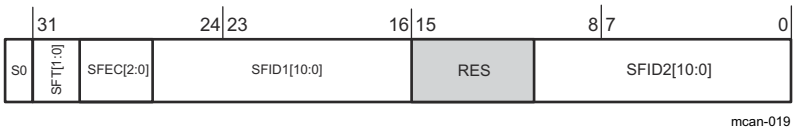


Figure 34-23. Standard Message ID Filter Element Structure

Table 34-13. Standard Message ID Filter Element Field Descriptions

Word	Bits	Field Name	Description
S0	31:30	SFT[1:0]	Standard Filter Type • 0x0: Range filter from SFID1 to SFID2 (SFID2 ≥ SFID1) • 0x1: Dual ID filter for SFID1 or SFID2 • 0x2: Classic filter: SFID1 = filter; SFID2 = mask • 0x3: Filter element disabled **Note:** With SFT = 11 the filter element is disabled and the acceptance filtering continues (same behavior as with SFEC = 000)
	29:27	SFEC[2:0]	Standard Filter Element Configuration All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = 100, 101, or 110 match sets interrupt flag MCAN_IR.HPM and, if enabled, an interrupt is generated. In this case, the MCAN_HPMS register is updated with the status of the priority match. • 0x0: Disable filter element • 0x1: Store in Rx FIFO 0 if filter matches • 0x2: Store in Rx FIFO 1 if filter matches • 0x3: Reject ID if filter matches • 0x4: Set priority if filter matches • 0x5: Set priority and store in FIFO 0 if filter matches • 0x6: Set priority and store in FIFO 1 if filter matches • 0x7: Store into Rx Buffer , configuration of SFT[1:0] ignored
	26:16	SFID1[10:0]	Standard Filter ID 1 When filtering for Rx buffers this field defines the ID of a standard message to be stored. The received identifiers must match exactly, no masking mechanism is used.
	15:11	RES	Reserved
	10:0	SFID2[10:0]	Standard Filter ID 2 This bit field has a different meaning depending on the configuration of SFEC: • SFEC = 001-110: Second ID of standard ID filter element • SFEC = 111: Filter for Rx buffers
		SFID2[10:9]	This filed is decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence. • 0x0: Store message into an Rx Buffer • 0x1: Debug Message A • 0x2: Debug Message B • 0x3: Debug Message C **Note:** Debug feature is not supported.
		SFID2[8:6]	This field is used to control the filter event pins at the Extension Interface. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one MCAN_ICKL period in case the filter matches. **Note:** Only two filter event pins are supported.
		SFID2[5:0]	This field defines the offset to the Rx Buffer Start Address MCAN_RXBC.RBSA field for storage of a matching message.

34.5.16.6 Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, the element address is the Filter List Extended Start Address MCAN_XIDFC.FLESA field plus two times the index of the filter element (0-63).

Figure 34-24 shows the Extended Message ID Filter element structure. Table 34-14 shows the Extended Message ID Filter element field descriptions.

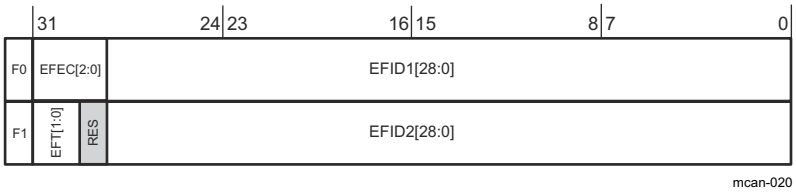


Figure 34-24. Extended Message ID Filter Element Structure

Table 34-14. Extended Message ID Filter Element Field Descriptions

Word	Bits	Field Name	Description
F0	31:29	EFEC[2:0]	Extended Filter Element Configuration All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = 100, 101, or 110 match sets interrupt flag MCAN_IR.HPM and, if enabled, an interrupt is generated. In this case, the MCAN_HPMS register is updated with the status of the priority match. • 0x0: Disable filter element • 0x1: Store in Rx FIFO 0 if filter matches • 0x2: Store in Rx FIFO 1 if filter matches • 0x3: Reject ID if filter matches • 0x4: Set priority if filter matches • 0x5: Set priority and store in FIFO 0 if filter matches • 0x6: Set priority and store in FIFO 1 if filter matches • 0x7: Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored
	28:0	EFID1[28:0]	Extended Filter ID 1 First ID of extended ID filter element. When filtering for Rx buffers this field defines the ID of an extended message to be stored. The received identifiers must match exactly, only XIDAM masking mechanism (see Section 34.5.13.1.5) is used.

Table 34-14. Extended Message ID Filter Element Field Descriptions (continued)

Word	Bits	Field Name	Description
F1	31:30	EFT[1:0]	Extended Filter Type • 0x0: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1) • 0x1: Dual ID filter for EFID1 or EFID2 • 0x2: Classic filter: EFID1 = filter, EFID2 = mask • 0x3: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1), XIDAM mask not applied
	29	RES	Reserved
	28:0	EFID2[28:0]	Extended Filter ID 2 This bit field has a different meaning depending on the configuration of EFEC: • EFEC = 001 - 110: Second ID of extended ID filter element • EFEC = 111: Filter for Rx buffers
		EFID2[10:9]	This field decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence. • 0x0: Store message into an Rx Buffer • 0x1: Debug Message A • 0x2: Debug Message B • 0x3: Debug Message C **Note:** Debug feature is not supported.
		EFID2[8:6]	This field is used to control the filter event pins at the Extension Interface. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one MCAN_ICKL period in case the filter matches. **Note:** Only two filter event pins are supported.
		EFID2[5:0]	This field defines the offset to the Rx Buffer Start Address MCAN_RXBC.RBSA field for storage of a matching message.

34.6 Software

34.6.1 MCAN Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/mcan

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

34.6.1.1 MCAN Loopback with Interrupts Example Using SYSCONFIG Tool - SINGLE_CORE

FILE: mcan_ex1_loopback_interrupts.c

This example illustrates the MCAN Loopback functionality. The internal loopback mode is entered. The message transmitted would be received by the node. The last address of memory is used for the Rx buffer. Peripheral configuration is done through SYSCONFIG

External Connections

- None.

Watch Variables

- error - Checks if there is an error that occurred when the data was sent using internal loopback.

34.6.1.2 MCAN Loopback with Polling Example Using SYSCONFIG Tool - SINGLE_CORE

FILE: mcan_ex2_loopback_polling.c

This example illustrates the MCAN Loopback functionality. The internal loopback mode is entered. The message transmitted would be received by the node. The last address of memory is used for the Rx buffer. Peripheral configuration is done through SYSCONFIG

External Connections

- None.

Watch Variables

- error - Checks if there is an error that occurred when the data was sent using internal loopback.

34.6.1.3 MCAN Loopback with Interrupts Example Using SYSCONFIG Tool - SINGLE_CORE

FILE: mcan_ex3_loopback_interrupts_fifo.c

This example illustrates the following MCAN features: -> Loopback mode -> Interrupt enabled -> TX fifo used -> Messages are transmitted in the order in which they are added to fifo -> RX fifo 0 used -> Multiple messages of same id can be received -> Rx range filtering is enabled with sfid1 = 0 and sfid2 = 4 Multiple messages (NUM_FRAMES) are transmitted and received -> Two messages have same id Peripheral configuration is done through SYSCONFIG

External Connections

- None.

Watch Variables

- error - Checks if there is an error that occurred when the data was sent using internal loopback.

34.7 MCAN Registers

This section describes the MCAN Registers.

34.7.1 MCAN Base Address Table

Table 34-15. MCAN Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
-	MCANA_DRIVER_BASE	0x6002_0000	YES	YES	YES	YES	YES	YES	-	YES
MCANSS_REGS	MCANSSA_BASE	0x6002_4000	YES	YES	YES	YES	YES	YES	-	YES
MCAN_REGS	MCANA_BASE	0x6002_4600	YES	YES	YES	YES	YES	YES	-	YES
MCAN_ERROR_REGS	MCANA_ERROR_BASE	0x6002_4800	YES	YES	YES	YES	YES	YES	-	YES
-	MCANB_DRIVER_BASE	0x6002_8000	YES	YES	YES	YES	YES	YES	-	YES
MCANSS_REGS	MCANBSS_BASE	0x6002_C000	YES	YES	YES	YES	YES	YES	-	YES
MCAN_REGS	MCANB_BASE	0x6002_C600	YES	YES	YES	YES	YES	YES	-	YES
MCAN_ERROR_REGS	MCANB_ERROR_BASE	0x6002_C800	YES	YES	YES	YES	YES	YES	-	YES
-	MCANC_DRIVER_BASE	0x6003_0000	YES	YES	YES	YES	YES	YES	-	YES
MCANSS_REGS	MCANCSS_BASE	0x6003_4000	YES	YES	YES	YES	YES	YES	-	YES
MCAN_REGS	MCANC_BASE	0x6003_4600	YES	YES	YES	YES	YES	YES	-	YES
MCAN_ERROR_REGS	MCANC_ERROR_BASE	0x6003_4800	YES	YES	YES	YES	YES	YES	-	YES
-	MCAND_DRIVER_BASE	0x6003_8000	YES	YES	YES	YES	YES	YES	-	YES
MCANSS_REGS	MCANDSS_BASE	0x6003_C000	YES	YES	YES	YES	YES	YES	-	YES
MCAN_REGS	MCAND_BASE	0x6003_C600	YES	YES	YES	YES	YES	YES	-	YES
MCAN_ERROR_REGS	MCAND_ERROR_BASE	0x6003_C800	YES	YES	YES	YES	YES	YES	-	YES
-	MCANE_DRIVER_BASE	0x6004_0000	YES	YES	YES	YES	YES	YES	-	YES
MCANSS_REGS	MCANESS_BASE	0x6004_4000	YES	YES	YES	YES	YES	YES	-	YES
MCAN_REGS	MCANE_BASE	0x6004_4600	YES	YES	YES	YES	YES	YES	-	YES
MCAN_ERROR_REGS	MCANE_ERROR_BASE	0x6004_4800	YES	YES	YES	YES	YES	YES	-	YES
-	MCANF_DRIVER_BASE	0x6004_8000	YES	YES	YES	YES	YES	YES	-	YES
MCANSS_REGS	MCANFSS_BASE	0x6004_C000	YES	YES	YES	YES	YES	YES	-	YES
MCAN_REGS	MCANF_BASE	0x6004_C600	YES	YES	YES	YES	YES	YES	-	YES
MCAN_ERROR_REGS	MCANF_ERROR_BASE	0x6004_C800	YES	YES	YES	YES	YES	YES	-	YES

34.7.2 MCANSS_REGS Registers

Table 34-16 lists the memory-mapped registers for the MCANSS_REGS registers. All register offset addresses not listed in Table 34-16 should be considered as reserved locations and the register contents should not be modified.

Table 34-16. MCANSS_REGS Registers

Offset	Acronym	Register Name	Protection
0h	MCANSS_PID	MCAN Subsystem Revision Register	
4h	MCANSS_CTRL	MCAN Subsystem Control Register	
8h	MCANSS_STAT	MCAN Subsystem Status Register	
Ch	MCANSS_ICS	MCAN Subsystem Interrupt Clear Shadow Register	
10h	MCANSS_IRS	MCAN Subsystem Interrupt Raw Satus Register	
14h	MCANSS_IECS	MCAN Subsystem Interrupt Enable Clear Shadow Register	
18h	MCANSS_IE	MCAN Subsystem Interrupt Enable Register	
1Ch	MCANSS_IES	MCAN Subsystem Interrupt Enable Status	
20h	MCANSS_EOI	MCAN Subsystem End of Interrupt	
24h	MCANSS_EXT_TS_PRESCALER	MCAN Subsystem External Timestamp Prescaler 0	
28h	MCANSS_EXT_TS_UNSERVICED_INTR_CNTR	MCAN Subsystem External Timestamp Unserviced Interrupts Counter	

Complex bit access types are encoded to fit into small table cells. Table 34-17 shows the codes that are used for access types in this section.

Table 34-17. MCANSS_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

34.7.2.1 MCANSS_PID Register (Offset = 0h) [Reset = 68E05101h]

MCANSS_PID is shown in Figure 34-25 and described in Table 34-18.

Return to the Summary Table.

MCAN Subsystem Revision Register

Figure 34-25. MCANSS_PID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCHEME		RESERVED		MODULE_ID											
R-1h		R-2h		R-8E0h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					MAJOR			RESERVED		MINOR					
R-Ah					R-1h			R-0h		R-1h					

Table 34-18. MCANSS_PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	PID Register Scheme Reset type: SYSRSn
29-28	RESERVED	R	2h	Reserved
27-16	MODULE_ID	R	8E0h	Module Identification Number Reset type: SYSRSn
15-11	RESERVED	R	Ah	Reserved
10-8	MAJOR	R	1h	Major Revision of the MCAN Subsystem Reset type: SYSRSn
7-6	RESERVED	R	0h	Reserved
5-0	MINOR	R	1h	Minor Revision of the MCAN Subsystem Reset type: SYSRSn

34.7.2.2 MCANSS_CTRL Register (Offset = 4h) [Reset = 00000008h]

MCANSS_CTRL is shown in Figure 34-26 and described in Table 34-19.

Return to the Summary Table.

MCAN Subsystem Control Register

Figure 34-26. MCANSS_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	EXT_TS_CNTR_EN	AUTOWAKEUP	WAKEUPREQEN	DBGSUSP_FREE	RESERVED		
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R-0h		

Table 34-19. MCANSS_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6	EXT_TS_CNTR_EN	R/W	0h	External Timestamp Counter Enable. 0 External timestamp counter disabled 1 External timestamp counter enabled Reset type: SYSRSn
5	AUTOWAKEUP	R/W	0h	Automatic Wakeup Enable. Enables the MCANSS to automatically clear the MCAN CCCR.INIT bit, fully waking the MCAN up, on an enabled wakeup request. 0 Disable the automatic write to CCCR.INIT 1 Enable the automatic write to CCCR.INIT Reset type: SYSRSn
4	WAKEUPREQEN	R/W	0h	Wakeup Request Enable. Enables the MCANSS to wakeup on CAN RXD activity. 0 Disable wakeup request 1 Enables wakeup request Reset type: SYSRSn
3	DBGSUSP_FREE	R/W	1h	Debug Suspend Free Bit. Enables debug suspend. 0 Disable debug suspend 1 Enable debug suspend Reset type: SYSRSn
2-0	RESERVED	R	0h	Reserved

34.7.2.3 MCANSS_STAT Register (Offset = 8h) [Reset = 0000000Xh]

MCANSS_STAT is shown in Figure 34-27 and described in Table 34-20.

Return to the Summary Table.

MCAN Subsystem Status Register

Figure 34-27. MCANSS_STAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					ENABLE_FDOE	MEM_INIT_DONE	RESET
R-0h					R-Xh	R-0h	R-0h

Table 34-20. MCANSS_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	ENABLE_FDOE	R	Xh	Flexible Datarate Operation Enable. Determines whether CAN FD operation may be enabled via the MCAN core CCCR.FDOE bit (bit 8) or if only standard CAN operation is possible with this instance of the MCAN. 0 MCAN is only capable of standard CAN communication 1 MCAN may be configured to perform CAN FD communication Reset type: SYSRSn
1	MEM_INIT_DONE	R	0h	Memory Initialization Done. 0 Message RAM initialization is in progress 1 Message RAM is initialized for use Reset type: SYSRSn
0	RESET	R	0h	Soft Reset Status. 0 Not in reset 1 Reset is in progress Reset type: SYSRSn

34.7.2.4 MCANSS_ICS Register (Offset = Ch) [Reset = 00000000h]

MCANSS_ICS is shown in Figure 34-28 and described in Table 34-21.

Return to the Summary Table.

MCAN Subsystem Interrupt Clear Shadow Register

Figure 34-28. MCANSS_ICS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							EXT_TS_CNTR_OVFL
R-0h							R-0/W1C-0h

Table 34-21. MCANSS_ICS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	EXT_TS_CNTR_OVFL	R-0/W1C	0h	External Timestamp Counter Overflow Interrupt Status Clear. Reads always return a 0. 0 Write of '0' has no effect 1 Write of '1' clears the MCANSS_IRS.EXT_TS_CNTR_OVFL bit Reset type: SYSRSn

34.7.2.5 MCANSS_IRS Register (Offset = 10h) [Reset = 00000000h]

MCANSS_IRS is shown in Figure 34-29 and described in Table 34-22.

Return to the Summary Table.

MCAN Subsystem Interrupt Raw Satus Register

Figure 34-29. MCANSS_IRS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							EXT_TS_CNTR_OVFL
R-0h							R/W1S-0h

Table 34-22. MCANSS_IRS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	EXT_TS_CNTR_OVFL	R/W1S	0h	External Timestamp Counter Overflow Interrupt Status. This bit is set by HW or by a SW write of '1'. To clear, use the MCANSS_ICS.EXT_TS_CNTR_OVFL bit. 0 External timestamp counter has not overflowed 1 External timestamp counter has overflowed When this bit is set to '1' by HW or SW, the MCANSS_EXT_TS_UNSERVICED_INTR_CNTR.EXT_TS_INTR_CNTR bit field will increment by 1. Reset type: SYSRSn

34.7.2.6 MCANSS_IECS Register (Offset = 14h) [Reset = 00000000h]

MCANSS_IECS is shown in Figure 34-30 and described in Table 34-23.

Return to the Summary Table.

MCAN Subsystem Interrupt Enable Clear Shadow Register

Figure 34-30. MCANSS_IECS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							EXT_TS_CNTR_OVFL
R-0h							R-0/W1C-0h

Table 34-23. MCANSS_IECS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	EXT_TS_CNTR_OVFL	R-0/W1C	0h	External Timestamp Counter Overflow Interrupt Enable Clear. Reads always return a 0. 0 Write of '0' has no effect 1 Write of '1' clears the MCANSS_IES.EXT_TS_CNTR_OVFL bit Reset type: SYSRSn

34.7.2.7 MCANSS_IE Register (Offset = 18h) [Reset = 00000000h]

MCANSS_IE is shown in Figure 34-31 and described in Table 34-24.

Return to the Summary Table.

MCAN Subsystem Interrupt Enable Register

Figure 34-31. MCANSS_IE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							EXT_TS_CNTR_OVFL
R-0h							R/W1S-0h

Table 34-24. MCANSS_IE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	EXT_TS_CNTR_OVFL	R/W1S	0h	External Timestamp Counter Overflow Interrupt Enable. A write of '0' has no effect. A write of '1' sets the MCANSS_IES.EXT_TS_CNTR_OVFL bit. Reset type: SYSRSn

34.7.2.8 MCANSS_IES Register (Offset = 1Ch) [Reset = 00000000h]

MCANSS_IES is shown in Figure 34-32 and described in Table 34-25.

Return to the Summary Table.

MCAN Subsystem Interrupt Enable Status

Figure 34-32. MCANSS_IES Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							EXT_TS_CNTR_OVFL
R-0h							R-0h

Table 34-25. MCANSS_IES Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	EXT_TS_CNTR_OVFL	R	0h	External Timestamp Counter Overflow Interrupt Enable Status. To set, use the CANSS_IE.EXT_TS_CNTR_OVFL bit. To clear, use the MCANSS_IECS.EXT_TS_CNTR_OVFL bit. 0 External timestamp counter overflow interrupt is not enabled 1 External timestamp counter overflow interrupt is enabled Reset type: SYSRSn

34.7.2.9 MCANSS_EOI Register (Offset = 20h) [Reset = 00000000h]

MCANSS_EOI is shown in Figure 34-33 and described in Table 34-26.

Return to the Summary Table.

MCAN Subsystem End of Interrupt

Figure 34-33. MCANSS_EOI Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EOI							
R-0h																								R-0/W1S-0h							

Table 34-26. MCANSS_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	EOI	R-0/W1S	0h	End of Interrupt. A write to this register will clear the associated interrupt. If the unserviced interrupt counter is > 1, another interrupt is generated. 0x00 External TS Interrupt is cleared 0x01 MCAN[0] interrupt is cleared 0x02 MCAN[1] interrupt is cleared Other writes are ignored. Reset type: SYSRSn

34.7.2.10 MCANSS_EXT_TS_PRESCALER Register (Offset = 24h) [Reset = 00000000h]

MCANSS_EXT_TS_PRESCALER is shown in Figure 34-34 and described in Table 34-27.

Return to the Summary Table.

MCAN Subsystem External Timestamp Prescaler 0

Figure 34-34. MCANSS_EXT_TS_PRESCALER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRESCALER																							
R-0h								R/W-0h																							

Table 34-27. MCANSS_EXT_TS_PRESCALER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	PRESCALER	R/W	0h	External Timestamp Prescaler Reload Value. The external timestamp count rate is the host (system) clock rate divided by this value, except in the case of 0. A zero value in this bit field will act identically to a value of 0x000001. Reset type: SYSRSn

34.7.2.11 MCANSS_EXT_TS_UNSERVICED_INTR_CNTR Register (Offset = 28h) [Reset = 00000000h]

MCANSS_EXT_TS_UNSERVICED_INTR_CNTR is shown in Figure 34-35 and described in Table 34-28.

Return to the Summary Table.

MCAN Subsystem External Timestamp Unserviced Interrupts Counter

Figure 34-35. MCANSS_EXT_TS_UNSERVICED_INTR_CNTR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			EXT_TS_INTR_CNTR				
R-0h			R-0h				

Table 34-28. MCANSS_EXT_TS_UNSERVICED_INTR_CNTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-0	EXT_TS_INTR_CNTR	R	0h	External Timestamp Counter Unserviced Rollover Interrupts. If this value is > 1, an MCANSS_EOI write of '1' to bit 0 will issue another interrupt. The status of this bit field is affected by the MCANSS_IRS.EXT_TS_CNTR_OVFL bit field. Reset type: SYSRSn

34.7.3 MCAN_REGS Registers

Table 34-29 lists the memory-mapped registers for the MCAN_REGS registers. All register offset addresses not listed in Table 34-29 should be considered as reserved locations and the register contents should not be modified.

Table 34-29. MCAN_REGS Registers

Offset	Acronym	Register Name	Protection
0h	MCAN_CREL	MCAN Core Release Register	
4h	MCAN_ENDN	MCAN Endian Register	
Ch	MCAN_DBTP	MCAN Data Bit Timing and Prescaler Register	
10h	MCAN_TEST	MCAN Test Register	
14h	MCAN_RWD	MCAN RAM Watchdog	
18h	MCAN_CCCR	MCAN CC Control Register	
1Ch	MCAN_NBTP	MCAN Nominal Bit Timing and Prescaler Register	
20h	MCAN_TSCC	MCAN Timestamp Counter Configuration	
24h	MCAN_TSCV	MCAN Timestamp Counter Value	
28h	MCAN_TOCC	MCAN Timeout Counter Configuration	
2Ch	MCAN_TOCV	MCAN Timeout Counter Value	
40h	MCAN_ECR	MCAN Error Counter Register	
44h	MCAN_PSR	MCAN Protocol Status Register	
48h	MCAN_TDCR	MCAN Transmitter Delay Compensation Register	
50h	MCAN_IR	MCAN Interrupt Register	
54h	MCAN_IE	MCAN Interrupt Enable	
58h	MCAN_ILS	MCAN Interrupt Line Select	
5Ch	MCAN_ILE	MCAN Interrupt Line Enable	
80h	MCAN_GFC	MCAN Global Filter Configuration	
84h	MCAN_SIDFC	MCAN Standard ID Filter Configuration	
88h	MCAN_XIDFC	MCAN Extended ID Filter Configuration	
90h	MCAN_XIDAM	MCAN Extended ID and Mask	
94h	MCAN_HPMS	MCAN High Priority Message Status	
98h	MCAN_NDAT1	MCAN New Data 1	
9Ch	MCAN_NDAT2	MCAN New Data 2	
A0h	MCAN_RXF0C	MCAN Rx FIFO 0 Configuration	
A4h	MCAN_RXF0S	MCAN Rx FIFO 0 Status	
A8h	MCAN_RXF0A	MCAN Rx FIFO 0 Acknowledge	
ACh	MCAN_RXBC	MCAN Rx Buffer Configuration	
B0h	MCAN_RXF1C	MCAN Rx FIFO 1 Configuration	
B4h	MCAN_RXF1S	MCAN Rx FIFO 1 Status	
B8h	MCAN_RXF1A	MCAN Rx FIFO 1 Acknowledge	
BCh	MCAN_RXESC	MCAN Rx Buffer / FIFO Element Size Configuration	
C0h	MCAN_TXBC	MCAN Tx Buffer Configuration	
C4h	MCAN_TXFQS	MCAN Tx FIFO / Queue Status	
C8h	MCAN_TXESC	MCAN Tx Buffer Element Size Configuration	
CCh	MCAN_TXBRP	MCAN Tx Buffer Request Pending	
D0h	MCAN_TXBAR	MCAN Tx Buffer Add Request	
D4h	MCAN_TXBCR	MCAN Tx Buffer Cancellation Request	
D8h	MCAN_TXBTO	MCAN Tx Buffer Transmission Occurred	

Table 34-29. MCAN_REGS Registers (continued)

Offset	Acronym	Register Name	Protection
DCh	MCAN_TXBCF	MCAN Tx Buffer Cancellation Finished	
E0h	MCAN_TXBTIE	MCAN Tx Buffer Transmission Interrupt Enable	
E4h	MCAN_TXBCIE	MCAN Tx Buffer Cancellation Finished Interrupt Enable	
F0h	MCAN_TXEFC	MCAN Tx Event FIFO Configuration	
F4h	MCAN_TXEFS	MCAN Tx Event FIFO Status	
F8h	MCAN_TXEFA	MCAN Tx Event FIFO Acknowledge	

Complex bit access types are encoded to fit into small table cells. Table 34-30 shows the codes that are used for access types in this section.

Table 34-30. MCAN_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
RC	R C	Read to Clear
RS	R S	Read to Set
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1SQ	W 1S Q	Write 1 to set Qualified. A condition must be met for this operation to occur.
WQ	W Q	Write Qualified. A condition must be met for this operation to occur.
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

34.7.3.1 MCAN_CREL Register (Offset = 0h) [Reset = 32380608h]

MCAN_CREL is shown in Figure 34-36 and described in Table 34-31.

Return to the Summary Table.

MCAN Core Release Register

Figure 34-36. MCAN_CREL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REL				STEP				SUBSTEP				YEAR			
R-3h				R-2h				R-3h				R-8h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MON								DAY							
R-6h								R-8h							

Table 34-31. MCAN_CREL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	REL	R	3h	Core Release. One digit, BCD-coded. Reset type: SYSRSn
27-24	STEP	R	2h	Step of Core Release. One digit, BCD-coded. Reset type: SYSRSn
23-20	SUBSTEP	R	3h	Sub-Step of Core Release. One digit, BCD-coded. Reset type: SYSRSn
19-16	YEAR	R	8h	Time Stamp Year. One digit, BCD-coded. Reset type: SYSRSn
15-8	MON	R	6h	Time Stamp Month. Two digits, BCD-coded. Reset type: SYSRSn
7-0	DAY	R	8h	Time Stamp Day. Two digits, BCD-coded. Reset type: SYSRSn

34.7.3.2 MCAN_ENDN Register (Offset = 4h) [Reset = 87654321h]

MCAN_ENDN is shown in Figure 34-37 and described in Table 34-32.

Return to the Summary Table.

MCAN Endian Register

Figure 34-37. MCAN_ENDN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETV																															
R-87654321h																															

Table 34-32. MCAN_ENDN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ETV	R	87654321h	Endianess Test Value. Reading the constant value maintained in this register allows software to determine the endianess of the host CPU. Reset type: SYSRSn

34.7.3.3 MCAN_DBTP Register (Offset = Ch) [Reset = 00000A33h]

MCAN_DBTP is shown in Figure 34-38 and described in Table 34-33.

Return to the Summary Table.

This register is only writable if bits CCCR.CCE and CCCR.INIT are set. The CAN bit time may be programed in the range of 4 to 49 time quanta. The CAN time quantum may be programmed in the range of 1 to 32 m_can_cclk periods. tq = (DBRP + 1) mtq.
DTSEG1 is the sum of Prop_Seg and Phase_Seg1. DTSEG2 is Phase_Seg2.
Therefore the length of the bit time is (programmed values) [DTSEG1 + DTSEG2 + 3] tq or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] tq.
The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.

Figure 34-38. MCAN_DBTP Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
TDC	RESERVED		DBRP				
R/WQ-0h	R-0h		R/WQ-0h				
15	14	13	12	11	10	9	8
RESERVED			DTSEG1				
R-0h			R/WQ-Ah				
7	6	5	4	3	2	1	0
DTSEG2				DSJW			
R/WQ-3h				R/WQ-3h			

Table 34-33. MCAN_DBTP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23	TDC	R/WQ	0h	Transmitter Delay Compensation 0 Transmitter Delay Compensation disabled 1 Transmitter Delay Compensation enabled +I107 Reset type: SYSRSn
22-21	RESERVED	R	0h	Reserved
20-16	DBRP	R/WQ	0h	Data Bit Rate Prescaler. The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
15-13	RESERVED	R	0h	Reserved
12-8	DTSEG1	R/WQ	Ah	Data Time Segment Before Sample Point. Valid values are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn

Table 34-33. MCAN_DBTP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-4	DTSEG2	R/WQ	3h	Data Time Segment After Sample Point. Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
3-0	DSJW	R/WQ	3h	Data Resynchronization Jump Width. Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn

34.7.3.4 MCAN_TEST Register (Offset = 10h) [Reset = 000000X0h]

MCAN_TEST is shown in Figure 34-39 and described in Table 34-34.

Return to the Summary Table.

Write access to the Test Register has to be enabled by setting bit CCCR.TEST to '1'. All Test Register functions are set to their reset values when bit CCCR.TEST is reset.
Loop Back Mode and software control of the internal CAN TX pin are hardware test modes. Programming of TX != '00' may disturb the message transfer on the CAN bus.

Figure 34-39. MCAN_TEST Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RX	TX		LBCK	RESERVED			
R-Xh	R/WQ-0h		R/WQ-0h	R-0h			

Table 34-34. MCAN_TEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	RX	R	Xh	Receive Pin. Monitors the actual value of the CAN receive pin. 0 The CAN bus is dominant (CAN RX pin = '0') 1 The CAN bus is recessive (CAN RX pin = '1') Reset type: SYSRSn
6-5	TX	R/WQ	0h	Control of Transmit Pin 00 CAN TX pin controlled by the CAN Core, updated at the end of the CAN bit time 01 Sample Point can be monitored at CAN TX pin 10 Dominant ('0') level at CAN TX pin 11 Recessive ('1') at CAN TX pin Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
4	LBCK	R/WQ	0h	Loop Back Mode 0 Reset value, Loop Back Mode is disabled 1 Loop Back Mode is enabled Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
3-0	RESERVED	R	0h	Reserved

34.7.3.5 MCAN_RWD Register (Offset = 14h) [Reset = 00000000h]

MCAN_RWD is shown in Figure 34-40 and described in Table 34-35.

Return to the Summary Table.

MCAN RAM Watchdog

Figure 34-40. MCAN_RWD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WDV								WDC							
R-0h																R-0h								R/WQ-0h							

Table 34-35. MCAN_RWD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	WDV	R	0h	Watchdog Value. Actual Message RAM Watchdog Counter Value. The RAM Watchdog monitors the READY output of the Message RAM. A Message RAM access via the MCAN's Generic Controller Interface starts the Message RAM Watchdog Counter with the value configured by the WDC field. The counter is reloaded with WDC when the Message RAM signals successful completion by activating its READY output. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt flag MCAN_IR.WDI is set. The RAM Watchdog Counter is clocked by the host (system) clock. Reset type: SYSRSn
7-0	WDC	R/WQ	0h	Watchdog Configuration. Start value of the Message RAM Watchdog Counter. With the reset value of '00' the counter is disabled. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn

34.7.3.6 MCAN_CCCR Register (Offset = 18h) [Reset = 00000001h]

MCAN_CCCR is shown in Figure 34-41 and described in Table 34-36.

Return to the Summary Table.

MCAN CC Control Register

Figure 34-41. MCAN_CCCR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
NISO	TXP	EFBI	PXHD	RESERVED		BRSE	FDOE
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R-0h		R/WQ-0h	R/WQ-0h
7	6	5	4	3	2	1	0
TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
R/W1SQ-0h	R/WQ-0h	R/W1SQ-0h	R/W-0h	R-0h	R/W1SQ-0h	R/WQ-0h	R/W-1h

Table 34-36. MCAN_CCCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	NISO	R/WQ	0h	Non ISO Operation. If this bit is set, the MCAN uses the CAN FD frame format as specified by the Bosch CAN FD Specification V1.0. 0 CAN FD frame format according to ISO 11898-1:2015 1 CAN FD frame format according to Bosch CAN FD Specification V1.0 Reset type: SYSRSn
14	TXP	R/WQ	0h	Transmit Pause. If this bit is set, the MCAN pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame. 0 Transmit pause disabled 1 Transmit pause enabled Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
13	EFBI	R/WQ	0h	Edge Filtering during Bus Integration 0 Edge filtering disabled 1 Two consecutive dominant tq required to detect an edge for hard synchronization Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
12	PXHD	R/WQ	0h	Protocol Exception Handling Disable 0 Protocol exception handling enabled 1 Protocol exception handling disabled Note: When protocol exception handling is disabled, the MCAN will transmit an error frame when it detects a protocol exception condition. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
11-10	RESERVED	R	0h	Reserved

Table 34-36. MCAN_CCCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	BRSE	R/WQ	0h	Bit Rate Switch Enable 0 Bit rate switching for transmissions disabled 1 Bit rate switching for transmissions enabled Note: When CAN FD operation is disabled FDOE = '0', BRSE is not evaluated. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
8	FDOE	R/WQ	0h	Flexible Datarate Operation Enable 0 FD operation disabled 1 FD operation enabled Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
7	TEST	R/W1SQ	0h	Test Mode Enable 0 Normal operation, register TEST holds reset values 1 Test Mode, write access to register TEST enabled Qualified Write 1 to Set is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
6	DAR	R/WQ	0h	Disable Automatic Retransmission 0 Automatic retransmission of messages not transmitted successfully enabled 1 Automatic retransmission disabled Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
5	MON	R/W1SQ	0h	Bus Monitoring Mode. Bit MON can only be set by SW when both CCE and INIT are set to '1'. The bit can be reset by SW at any time. 0 Bus Monitoring Mode is disabled 1 Bus Monitoring Mode is enabled Qualified Write 1 to Set is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
4	CSR	R/W	0h	Clock Stop Request 0 No clock stop is requested 1 Clock stop requested. When clock stop is requested, first INIT and then CSA will be set after all pending transfer requests have been completed and the CAN bus reached idle. Reset type: SYSRSn
3	CSA	R	0h	Clock Stop Acknowledge 0 No clock stop acknowledged 1 MCAN may be set in power down by stopping the Host and CAN clocks Reset type: SYSRSn
2	ASM	R/W1SQ	0h	Restricted Operation Mode. Bit ASM can only be set by SW when both CCE and INIT are set to '1'. The bit can be reset by SW at any time. 0 Normal CAN operation 1 Restricted Operation Mode active Qualified Write 1 to Set is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn

Table 34-36. MCAN_CCCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	CCE	R/WQ	0h	Configuration Change Enable 0 The CPU has no write access to the protected configuration registers 1 The CPU has write access to the protected configuration registers (while CCCR.INIT = '1') Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
0	INIT	R/W	1h	Initialization 0 Normal Operation 1 Initialization is started Note: Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to INIT can be read back. Therefore the programmer has to assure that the previous value written to INIT has been accepted by reading INIT before setting INIT to a new value. Reset type: SYSRSn

34.7.3.7 MCAN_NBTP Register (Offset = 1Ch) [Reset = 06000A03h]

MCAN_NBTP is shown in Figure 34-42 and described in Table 34-37.

Return to the Summary Table.

This register is only writable if bits CCCR.CCE and CCCR.INIT are set. The CAN bit time may be programed in the range of 4 to 385 time quanta. The CAN time quantum may be programmed in the range of 1 to 512 m_can_cclk periods. tq = (NBRP + 1) mtq.
NTSEG1 is the sum of Prop_Seg and Phase_Seg1. NTSEG2 is Phase_Seg2.
Therefore the length of the bit time is (programmed values) [NTSEG1 + NTSEG2 + 3] tq or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] tq.
The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.
Note: With a CAN clock of 8 MHz, the reset value of 0x06000A03 configures the MCAN for a bit rate of 500 kBit/s.

Figure 34-42. MCAN_NBTP Register

31	30	29	28	27	26	25	24
NSJW							NBRP
R/WQ-3h							R/WQ-0h
23	22	21	20	19	18	17	16
NBRP							
R/WQ-0h							
15	14	13	12	11	10	9	8
NTSEG1							
R/WQ-Ah							
7	6	5	4	3	2	1	0
RESERVED	NTSEG2						
R-0h	R/WQ-3h						

Table 34-37. MCAN_NBTP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NSJW	R/WQ	3h	Nominal (Re)Synchronization Jump Width. Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
24-16	NBRP	R/WQ	0h	Nominal Bit Rate Prescaler. The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 511. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
15-8	NTSEG1	R/WQ	Ah	Nominal Time Segment Before Sample Point. Valid values are 1 to 255. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
7	RESERVED	R	0h	Reserved

Table 34-37. MCAN_NBTP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6-0	NTSEG2	R/WQ	3h	Nominal Time Segment After Sample Point. Valid values are 1 to 127. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn

34.7.3.8 MCAN_TSCC Register (Offset = 20h) [Reset = 00000000h]

MCAN_TSCC is shown in Figure 34-43 and described in Table 34-38.

Return to the Summary Table.

MCAN Timestamp Counter Configuration

Figure 34-43. MCAN_TSCC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED												TCP			
R-0h												R/WQ-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														TSS	
R-0h														R/WQ-0h	

Table 34-38. MCAN_TSCC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19-16	TCP	R/WQ	0h	Timestamp Counter Prescaler. Configures the timestamp and timeout counters time unit in multiples of CAN bit times. Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. Note: With CAN FD an external counter is required for timestamp generation (TSS = '10'). Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	TSS	R/WQ	0h	Timestamp Select 00 Timestamp counter value always 0x0000 01 Timestamp counter value incremented according to TCP 10 External timestamp counter value used 11 Same as '00' Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn

34.7.3.9 MCAN_TSCV Register (Offset = 24h) [Reset = 00000000h]

MCAN_TSCV is shown in Figure 34-44 and described in Table 34-39.

Return to the Summary Table.

MCAN Timestamp Counter Value

Figure 34-44. MCAN_TSCV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TSC															
R-0h																R/W-0h															

Table 34-39. MCAN_TSCV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	TSC	R/W	0h	Timestamp Counter. The internal/external Timestamp Counter value is captured on start of frame (both Rx and Tx). When TSCC.TSS = '01', the Timestamp Counter is incremented in multiples of CAN bit times, [1...16], depending on the configuration of TSCC.TCP. A wrap around sets interrupt flag IR.TSW. Write access resets the counter to zero. When TSCC.TSS = '10', TSC reflects the External Timestamp Counter value, and a write access has no impact. Note: A 'wrap around' is a change of the Timestamp Counter value from non-zero to zero not caused by write access to MCAN_TSCV. Reset type: SYSRSn

34.7.3.10 MCAN_TOCC Register (Offset = 28h) [Reset = FFFF0000h]

MCAN_TOCC is shown in Figure 34-45 and described in Table 34-40.

Return to the Summary Table.

MCAN Timeout Counter Configuration

Figure 34-45. MCAN_TOCC Register

31	30	29	28	27	26	25	24
TOP							
R/WQ-FFFFh							
23	22	21	20	19	18	17	16
TOP							
R/WQ-FFFFh							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					TOS		ETOC
R-0h					R/WQ-0h		R/WQ-0h

Table 34-40. MCAN_TOCC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TOP	R/WQ	FFFFh	Timeout Period. Start value of the Timeout Counter (down-counter). Configures the Timeout Period. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
15-3	RESERVED	R	0h	Reserved
2-1	TOS	R/WQ	0h	Timeout Select. When operating in Continuous mode, a write to TOCV presets the counter to the value configured by TOCC.TOP and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOCC.TOP. Down-counting is started when the first FIFO element is stored. 00 Continuous operation 01 Timeout controlled by Tx Event FIFO 10 Timeout controlled by Rx FIFO 0 11 Timeout controlled by Rx FIFO 1 Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
0	ETOC	R/WQ	0h	Enable Timeout Counter 0 Timeout Counter disabled 1 Timeout Counter enabled Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn

34.7.3.11 MCAN_TOCV Register (Offset = 2Ch) [Reset = 0000FFFFh]

MCAN_TOCV is shown in Figure 34-46 and described in Table 34-41.

Return to the Summary Table.

MCAN Timeout Counter Value

Figure 34-46. MCAN_TOCV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TOC															
R-0h																R/W-FFFFh															

Table 34-41. MCAN_TOCV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	TOC	R/W	FFFFh	Timeout Counter. The Timeout Counter is decremented in multiples of CAN bit times, [1...16], depending on the configuration of TSCC.TCP. When decremented to zero, interrupt flag IR.TOO is set and the Timeout Counter is stopped. Start and reset/restart conditions are configured via TOCC.TOS. Reset type: SYSRSn

34.7.3.12 MCAN_ECR Register (Offset = 40h) [Reset = 00000000h]

MCAN_ECR is shown in Figure 34-47 and described in Table 34-42.

Return to the Summary Table.

MCAN Error Counter Register

Figure 34-47. MCAN_ECR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								CEL							
R-0h								RC-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RP	REC							TEC							
R-0h	R-0h							R-0h							

Table 34-42. MCAN_ECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	CEL	RC	0h	CAN Error Logging. The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or the Receive Error Counter to be incremented. It is reset by read access to CEL. The counter stops at 0xFF the next increment of TEC or REC sets interrupt flag IR.ELO. Note: When CCCR.ASM is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented. Reset type: SYSRSn
15	RP	R	0h	Receive Error Passive 0 The Receive Error Counter is below the error passive level of 128 1 The Receive Error Counter has reached the error passive level of 128 Reset type: SYSRSn
14-8	REC	R	0h	Receive Error Counter. Actual state of the Receive Error Counter, values between 0 and 127. Note: When CCCR.ASM is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented. Reset type: SYSRSn
7-0	TEC	R	0h	Transmit Error Counter. Actual state of the Transmit Error Counter, values between 0 and 255. Note: When CCCR.ASM is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented. Reset type: SYSRSn

34.7.3.13 MCAN_PSR Register (Offset = 44h) [Reset = 00000707h]

MCAN_PSR is shown in Figure 34-48 and described in Table 34-43.

Return to the Summary Table.

MCAN Protocol Status Register

Figure 34-48. MCAN_PSR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED	TDCV						
R-0h	R-0h						
15	14	13	12	11	10	9	8
RESERVED	PXE	RFDF	RBRS	RESI	DLEC		
R-0h	RC-0h	RC-0h	RC-0h	RC-0h	RS-7h		
7	6	5	4	3	2	1	0
BO	EW	EP	ACT		LEC		
R-0h	R-0h	R-0h	R-0h		RS-7h		

Table 34-43. MCAN_PSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	0h	Reserved
22-16	TDCV	R	0h	Transmitter Delay Compensation Value. Position of the secondary sample point, defined by the sum of the measured delay from the internal CAN TX signal to the internal CAN RX signal and TDCR.TDCO. The SSP position is, in the data phase, the number of mtq between the start of the transmitted bit and the secondary sample point. Valid values are 0 to 127 mtq. Reset type: SYSRSn
15	RESERVED	R	0h	Reserved
14	PXE	RC	0h	Protocol Exception Event 0 No protocol exception event occurred since last read access 1 Protocol exception event occurred Reset type: SYSRSn
13	RFDF	RC	0h	Received a CAN FD Message. This bit is set independent of acceptance filtering. 0 Since this bit was reset by the CPU, no CAN FD message has been received 1 Message in CAN FD format with FDF flag set has been received Reset type: SYSRSn
12	RBRS	RC	0h	BRS Flag of Last Received CAN FD Message. This bit is set together with RFDF, independent of acceptance filtering. 0 Last received CAN FD message did not have its BRS flag set 1 Last received CAN FD message had its BRS flag set Reset type: SYSRSn
11	RESI	RC	0h	ESI Flag of Last Received CAN FD Message. This bit is set together with RFDF, independent of acceptance filtering. 0 Last received CAN FD message did not have its ESI flag set 1 Last received CAN FD message had its ESI flag set Reset type: SYSRSn

Table 34-43. MCAN_PSR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-8	DLEC	RS	7h	Data Phase Last Error Code. Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for LEC. This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error. Reset type: SYSRSn
7	BO	R	0h	Bus_Off Status 0 The M_CAN is not Bus_Off 1 The M_CAN is in Bus_Off state Reset type: SYSRSn
6	EW	R	0h	Warning Status 0 Both error counters are below the Error_Warning limit of 96 1 At least one of error counter has reached the Error_Warning limit of 96 Reset type: SYSRSn
5	EP	R	0h	Error Passive 0 The M_CAN is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected 1 The M_CAN is in the Error_Passive state Reset type: SYSRSn
4-3	ACT	R	0h	Node Activity. Monitors the module's CAN communication state. 00 Synchronizing - node is synchronizing on CAN communication 01 Idle - node is neither receiver nor transmitter 10 Receiver - node is operating as receiver 11 Transmitter - node is operating as transmitter Note: ACT is set to '00' by a Protocol Exception Event. Reset type: SYSRSn

Table 34-43. MCAN_PSR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	LEC	RS	7h	Last Error Code. The LEC indicates the type of the last error to occur on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error. 0 No Error: No error occurred since LEC has been reset by successful reception or transmission. 1 Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed. 2 Form Error: A fixed format part of a received frame has the wrong format. 3 AckError: The message transmitted by the MCAN was not acknowledged by another node. 4 Bit1Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus value was dominant. 5 Bit0Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value '0'), but the monitored bus value was recessive. During Bus_Off recovery this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed). 6 CRCError: The CRC check sum of a received message was incorrect. The CRC of an incoming message does not match with the CRC calculated from the received data. 7 NoChange: Any read access to the Protocol Status Register re-initializes the LEC to '7'. When the LEC shows the value '7', no CAN bus event was detected since the last CPU read access to the Protocol Status Register. Note: When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in DLEC instead of LEC. An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error. Note: The Bus_Off recovery sequence (see ISO 11898-1:2015) cannot be shortened by setting or resetting CCCR.INIT. If the device goes Bus_Off, it will set CCCR.INIT of its own accord, stopping all bus activities. Once CCCR.INIT has been cleared by the CPU, the device will then wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus_Off recovery sequence, the Error Management Counters will be reset. During the waiting time after the resetting of CCCR.INIT, each time a sequence of 11 recessive bits has been monitored, a Bit0Error code is written to PSR.LEC, enabling the CPU to readily check up whether the CAN bus is stuck at dominant or continuously disturbed and to monitor the Bus_Off recovery sequence. ECR.REC is used to count these sequences. Reset type: SYSRSn

34.7.3.14 MCAN_TDCR Register (Offset = 48h) [Reset = 00000000h]

MCAN_TDCR is shown in Figure 34-49 and described in Table 34-44.

Return to the Summary Table.

MCAN Transmitter Delay Compensation Register

Figure 34-49. MCAN_TDCR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	TDCO						
R-0h	R/WQ-0h						
7	6	5	4	3	2	1	0
RESERVED	TDCF						
R-0h	R/WQ-0h						

Table 34-44. MCAN_TDCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RESERVED	R	0h	Reserved
14-8	TDCO	R/WQ	0h	Transmitter Delay Compensation Offset. Offset value defining the distance between the measured delay from the internal CAN TX signal to the internal CAN RX signal and the secondary sample point. Valid values are 0 to 127 mtq. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6-0	TDCF	R/WQ	0h	Transmitter Delay Compensation Filter Window Length. Defines the minimum value for the SSP position, dominant edges on the internal CAN RX signal that would result in an earlier SSP position are ignored for transmitter delay measurement. The feature is enabled when TDCF is configured to a value greater than TDCO. Valid values are 0 to 127 mtq. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn

34.7.3.15 MCAN_IR Register (Offset = 50h) [Reset = 80000000h]

MCAN_IR is shown in Figure 34-50 and described in Table 34-45.

Return to the Summary Table.

The flags are set when one of the listed conditions is detected (edge-sensitive). The flags remain set until the Host clears them. Aflag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. Ahard reset will clear the register. The configuration of IE controls whether an interrupt is generated. The configuration of ILS controls on which interrupt line an interrupt is signalled.

Figure 34-50. MCAN_IR Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	ARA	PED	PEA	WDI	BO	EW
R-1h	R-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	16
EP	ELO	BEU	RESERVED	DRX	TOO	MRAF	TSW
R/W1C-0h	R/W1C-0h	R/W1C-0h	R-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
15	14	13	12	11	10	9	8
TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

Table 34-45. MCAN_IR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	1h	Reserved
30	RESERVED	R	0h	Reserved
29	ARA	R/W1C	0h	Access to Reserved Address 0 No access to reserved address occurred 1 Access to reserved address occurred Reset type: SYSRSn
28	PED	R/W1C	0h	Protocol Error in Data Phase (Data Bit Time is used) 0 No protocol error in data phase 1 Protocol error in data phase detected (PSR.DLEC != 0,7) Reset type: SYSRSn
27	PEA	R/W1C	0h	Protocol Error in Arbitration Phase (Nominal Bit Time is used) 0 No protocol error in arbitration phase 1 Protocol error in arbitration phase detected (PSR.LEC != 0,7) Reset type: SYSRSn
26	WDI	R/W1C	0h	Watchdog Interrupt 0 No Message RAM Watchdog event occurred 1 Message RAM Watchdog event due to missing READY Reset type: SYSRSn
25	BO	R/W1C	0h	Bus_Off Status 0 Bus_Off status unchanged 1 Bus_Off status changed Reset type: SYSRSn
24	EW	R/W1C	0h	Warning Status 0 Error_Warning status unchanged 1 Error_Warning status changed Reset type: SYSRSn

Table 34-45. MCAN_IR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	EP	R/W1C	0h	Error Passive 0 Error_Passive status unchanged 1 Error_Passive status changed Reset type: SYSRSn
22	ELO	R/W1C	0h	Error Logging Overflow 0 CAN Error Logging Counter did not overflow 1 Overflow of CAN Error Logging Counter occurred Reset type: SYSRSn
21	BEU	R/W1C	0h	Bit Error Uncorrected. Message RAM bit error detected, uncorrected. This bit is set when a double bit error is detected by the ECC aggregator attached to the Message RAM. An uncorrected Message RAM bit error sets CCCR.INIT to '1'. This is done to avoid transmission of corrupted data. 0 No bit error detected when reading from Message RAM 1 Bit error detected, uncorrected (e.g. parity logic) Reset type: SYSRSn
20	RESERVED	R	0h	Reserved
19	DRX	R/W1C	0h	Message Stored to Dedicated Rx Buffer. The flag is set whenever a received message has been stored into a dedicated Rx Buffer. 0 No Rx Buffer updated 1 At least one received message stored into an Rx Buffer Reset type: SYSRSn
18	TOO	R/W1C	0h	Timeout Occurred 0 No timeout 1 Timeout reached Reset type: SYSRSn
17	MRAF	R/W1C	0h	Message RAM Access Failure. The flag is set, when the Rx Handler: - has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message. - was not able to write a message to the Message RAM. In this case message storage is aborted. In both cases the FIFO put index is not updated resp. the New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location. The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the MCAN is switched into Restricted Operation Mode. To leave Restricted Operation Mode, the Host CPU has to reset CCCR.ASM. 0 No Message RAM access failure occurred 1 Message RAM access failure occurred Reset type: SYSRSn
16	TSW	R/W1C	0h	Timestamp Wraparound 0 No timestamp counter wrap-around 1 Timestamp counter wrapped around Reset type: SYSRSn
15	TEFL	R/W1C	0h	Tx Event FIFO Element Lost 0 No Tx Event FIFO element lost 1 Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero Reset type: SYSRSn
14	TEFF	R/W1C	0h	Tx Event FIFO Full 0 Tx Event FIFO not full 1 Tx Event FIFO full Reset type: SYSRSn

Table 34-45. MCAN_IR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	TEFW	R/W1C	0h	Tx Event FIFO Watermark Reached 0 Tx Event FIFO fill level below watermark 1 Tx Event FIFO fill level reached watermark Reset type: SYSRSn
12	TEFN	R/W1C	0h	Tx Event FIFO New Entry 0 Tx Event FIFO unchanged 1 Tx Handler wrote Tx Event FIFO element Reset type: SYSRSn
11	TFE	R/W1C	0h	Tx FIFO Empty 0 Tx FIFO non-empty 1 Tx FIFO empty Reset type: SYSRSn
10	TCF	R/W1C	0h	Transmission Cancellation Finished 0 No transmission cancellation finished 1 Transmission cancellation finished Reset type: SYSRSn
9	TC	R/W1C	0h	Transmission Completed 0 No transmission completed 1 Transmission completed Reset type: SYSRSn
8	HPM	R/W1C	0h	High Priority Message 0 No high priority message received 1 High priority message received Reset type: SYSRSn
7	RF1L	R/W1C	0h	Rx FIFO 1 Message Lost 0 No Rx FIFO 1 message lost 1 Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero Reset type: SYSRSn
6	RF1F	R/W1C	0h	Rx FIFO 1 Full 0 Rx FIFO 1 not full 1 Rx FIFO 1 full Reset type: SYSRSn
5	RF1W	R/W1C	0h	Rx FIFO 1 Watermark Reached 0 Rx FIFO 1 fill level below watermark 1 Rx FIFO 1 fill level reached watermark Reset type: SYSRSn
4	RF1N	R/W1C	0h	Rx FIFO 1 New Message 0 No new message written to Rx FIFO 1 1 New message written to Rx FIFO 1 Reset type: SYSRSn
3	RF0L	R/W1C	0h	Rx FIFO 0 Message Lost 0 No Rx FIFO 0 message lost 1 Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero Reset type: SYSRSn
2	RF0F	R/W1C	0h	Rx FIFO 0 Full 0 Rx FIFO 0 not full 1 Rx FIFO 0 full Reset type: SYSRSn
1	RF0W	R/W1C	0h	Rx FIFO 0 Watermark Reached 0 Rx FIFO 0 fill level below watermark 1 Rx FIFO 0 fill level reached watermark Reset type: SYSRSn

Table 34-45. MCAN_IR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	RF0N	R/W1C	0h	Rx FIFO 0 New Message 0 No new message written to Rx FIFO 0 1 New message written to Rx FIFO 0 Reset type: SYSRSn

34.7.3.16 MCAN_IE Register (Offset = 54h) [Reset = 00000000h]

MCAN_IE is shown in Figure 34-51 and described in Table 34-46.

Return to the Summary Table.

MCAN Interrupt Enable

Figure 34-51. MCAN_IE Register

31	30	29	28	27	26	25	24
RESERVED		ARAE	PEDE	PEAE	WDIE	BOE	EWE
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 34-46. MCAN_IE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Reserved
29	ARAE	R/W	0h	Access to Reserved Address Enable Reset type: SYSRSn
28	PEDE	R/W	0h	Protocol Error in Data Phase Enable Reset type: SYSRSn
27	PEAE	R/W	0h	Protocol Error in Arbitration Phase Enable Reset type: SYSRSn
26	WDIE	R/W	0h	Watchdog Interrupt Enable Reset type: SYSRSn
25	BOE	R/W	0h	Bus_Off Status Enable Reset type: SYSRSn
24	EWE	R/W	0h	Warning Status Enable Reset type: SYSRSn
23	EPE	R/W	0h	Error Passive Enable Reset type: SYSRSn
22	ELOE	R/W	0h	Error Logging Overflow Enable Reset type: SYSRSn
21	BEUE	R/W	0h	Bit Error Uncorrected Enable Reset type: SYSRSn
20	BECE	R/W	0h	Bit Error Corrected Enable A separate interrupt line reserved for corrected bit errors is provided via the MCAN_ERROR_REGS. It advised for the user to use these registers and leave this bit cleared to '0'. Reset type: SYSRSn
19	DRXE	R/W	0h	Message Stored to Dedicated Rx Buffer Enable Reset type: SYSRSn
18	TOOE	R/W	0h	Timeout Occurred Enable Reset type: SYSRSn

Table 34-46. MCAN_IE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	MRAFE	R/W	0h	Message RAM Access Failure Enable Reset type: SYSRSn
16	TSWE	R/W	0h	Timestamp Wraparound Enable Reset type: SYSRSn
15	TEFLE	R/W	0h	Tx Event FIFO Element Lost Enable Reset type: SYSRSn
14	TEFFE	R/W	0h	Tx Event FIFO Full Enable Reset type: SYSRSn
13	TEFWE	R/W	0h	Tx Event FIFO Watermark Reached Enable Reset type: SYSRSn
12	TEFNE	R/W	0h	Tx Event FIFO New Entry Enable Reset type: SYSRSn
11	TFEE	R/W	0h	Tx FIFO Empty Enable Reset type: SYSRSn
10	TCFE	R/W	0h	Transmission Cancellation Finished Enable Reset type: SYSRSn
9	TCE	R/W	0h	Transmission Completed Enable Reset type: SYSRSn
8	HPME	R/W	0h	High Priority Message Enable Reset type: SYSRSn
7	RF1LE	R/W	0h	Rx FIFO 1 Message Lost Enable Reset type: SYSRSn
6	RF1FE	R/W	0h	Rx FIFO 1 Full Enable Reset type: SYSRSn
5	RF1WE	R/W	0h	Rx FIFO 1 Watermark Reached Enable Reset type: SYSRSn
4	RF1NE	R/W	0h	Rx FIFO 1 New Message Enable Reset type: SYSRSn
3	RF0LE	R/W	0h	Rx FIFO 0 Message Lost Enable Reset type: SYSRSn
2	RF0FE	R/W	0h	Rx FIFO 0 Full Enable Reset type: SYSRSn
1	RF0WE	R/W	0h	Rx FIFO 0 Watermark Reached Enable Reset type: SYSRSn
0	RF0NE	R/W	0h	Rx FIFO 0 New Message Enable Reset type: SYSRSn

34.7.3.17 MCAN_ILS Register (Offset = 58h) [Reset = 00000000h]

MCAN_ILS is shown in Figure 34-52 and described in Table 34-47.

Return to the Summary Table.

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from the Interrupt Register to one of the two module interrupt lines. For interrupt generation the respective interrupt line has to be enabled via ILE.EINT0 and ILE.EINT1.

Figure 34-52. MCAN_ILS Register

31	30	29	28	27	26	25	24
RESERVED		ARAL	PEDL	PEAL	WDIL	BOL	EWL
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 34-47. MCAN_ILS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Reserved
29	ARAL	R/W	0h	Access to Reserved Address Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
28	PEDL	R/W	0h	Protocol Error in Data Phase Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
27	PEAL	R/W	0h	Protocol Error in Arbitration Phase Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
26	WDIL	R/W	0h	Watchdog Interrupt Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
25	BOL	R/W	0h	Bus_Off Status Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
24	EWL	R/W	0h	Warning Status Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
23	EPL	R/W	0h	Error Passive Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn

Table 34-47. MCAN_ILS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	ELOL	R/W	0h	Error Logging Overflow Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
21	BEUL	R/W	0h	Bit Error Uncorrected Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
20	BECL	R/W	0h	Bit Error Corrected Line A separate interrupt line reserved for corrected bit errors is provided via the MCAN_ERROR_REGS. It advised for the user to use these registers and leave the MCAN_IE.BECE bit cleared to '0' (disabled), thereby relegating this bit to not applicable. Reset type: SYSRSn
19	DRXL	R/W	0h	Message Stored to Dedicated Rx Buffer Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
18	TOOL	R/W	0h	Timeout Occurred Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
17	MRAFL	R/W	0h	Message RAM Access Failure Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
16	TSWL	R/W	0h	Timestamp Wraparound Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
15	TEFLL	R/W	0h	Tx Event FIFO Element Lost Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
14	TEFFL	R/W	0h	Tx Event FIFO Full Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
13	TEFWL	R/W	0h	Tx Event FIFO Watermark Reached Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
12	TEFNL	R/W	0h	Tx Event FIFO New Entry Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
11	TFEL	R/W	0h	Tx FIFO Empty Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
10	TCFL	R/W	0h	Transmission Cancellation Finished Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn

Table 34-47. MCAN_ILS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	TCL	R/W	0h	Transmission Completed Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
8	HPML	R/W	0h	High Priority Message Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
7	RF1LL	R/W	0h	Rx FIFO 1 Message Lost Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
6	RF1FL	R/W	0h	Rx FIFO 1 Full Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
5	RF1WL	R/W	0h	Rx FIFO 1 Watermark Reached Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
4	RF1NL	R/W	0h	Rx FIFO 1 New Message Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
3	RF0LL	R/W	0h	Rx FIFO 0 Message Lost Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
2	RF0FL	R/W	0h	Rx FIFO 0 Full Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
1	RF0WL	R/W	0h	Rx FIFO 0 Watermark Reached Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn
0	RF0NL	R/W	0h	Rx FIFO 0 New Message Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1 Reset type: SYSRSn

34.7.3.18 MCAN_ILE Register (Offset = 5Ch) [Reset = 00000000h]

MCAN_ILE is shown in Figure 34-53 and described in Table 34-48.

Return to the Summary Table.

MCAN Interrupt Line Enable

Figure 34-53. MCAN_ILE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						EINT1	EINT0
R-0h						R/W-0h	R/W-0h

Table 34-48. MCAN_ILE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	EINT1	R/W	0h	Enable Interrupt Line 1 0 Interrupt Line 1 is disabled 1 Interrupt Line 1 is enabled Reset type: SYSRSn
0	EINT0	R/W	0h	Enable Interrupt Line 0 0 Interrupt Line 0 is disabled 1 Interrupt Line 0 is enabled Reset type: SYSRSn

34.7.3.19 MCAN_GFC Register (Offset = 80h) [Reset = 00000000h]

MCAN_GFC is shown in Figure 34-54 and described in Table 34-49.

Return to the Summary Table.

MCAN Global Filter Configuration

Figure 34-54. MCAN_GFC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		ANFS		ANFE		RRFS	RRFE
R-0h		R/WQ-0h		R/WQ-0h		R/WQ-0h	R/WQ-0h

Table 34-49. MCAN_GFC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-4	ANFS	R/WQ	0h	Accept Non-matching Frames Standard. Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated. 00 Accept in Rx FIFO 0 01 Accept in Rx FIFO 1 10 Reject 11 Reject Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
3-2	ANFE	R/WQ	0h	Accept Non-matching Frames Extended. Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated. 00 Accept in Rx FIFO 0 01 Accept in Rx FIFO 1 10 Reject 11 Reject Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
1	RRFS	R/WQ	0h	Reject Remote Frames Standard 0 Filter remote frames with 11-bit standard IDs 1 Reject all remote frames with 11-bit standard IDs Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
0	RRFE	R/WQ	0h	Reject Remote Frames Extended 0 Filter remote frames with 29-bit extended IDs 1 Reject all remote frames with 29-bit extended IDs Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn

34.7.3.20 MCAN_SIDFC Register (Offset = 84h) [Reset = 00000000h]

MCAN_SIDFC is shown in Figure 34-55 and described in Table 34-50.

Return to the Summary Table.

MCAN Standard ID Filter Configuration

Figure 34-55. MCAN_SIDFC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
LSS							
R/WQ-0h							
15	14	13	12	11	10	9	8
FLSSA							
R/WQ-0h							
7	6	5	4	3	2	1	0
FLSSA						RESERVED	
R/WQ-0h						R-0h	

Table 34-50. MCAN_SIDFC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	LSS	R/WQ	0h	List Size Standard 0 No standard Message ID filter 1-128 Number of standard Message ID filter elements >128 Values greater than 128 are interpreted as 128 Reset type: SYSRSn
15-2	FLSSA	R/WQ	0h	Filter List Standard Start Address. Start address of standard Message ID filter list (32-bit word address). Reset type: SYSRSn
1-0	RESERVED	R	0h	Reserved

34.7.3.21 MCAN_XIDFC Register (Offset = 88h) [Reset = 00000000h]

MCAN_XIDFC is shown in Figure 34-56 and described in Table 34-51.

Return to the Summary Table.

MCAN Extended ID Filter Configuration

Figure 34-56. MCAN_XIDFC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED	LSE						
R-0h	R/WQ-0h						
15	14	13	12	11	10	9	8
FLESA							
R/WQ-0h							
7	6	5	4	3	2	1	0
FLESA						RESERVED	
R/WQ-0h						R-0h	

Table 34-51. MCAN_XIDFC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	0h	Reserved
22-16	LSE	R/WQ	0h	List Size Extended 0 No extended Message ID filter 1-64 Number of extended Message ID filter elements >64 Values greater than 64 are interpreted as 64 Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
15-2	FLESA	R/WQ	0h	Filter List Extended Start Address. Start address of extended Message ID filter list (32-bit word address). Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
1-0	RESERVED	R	0h	Reserved

34.7.3.22 MCAN_XIDAM Register (Offset = 90h) [Reset = 1FFFFFFFh]

MCAN_XIDAM is shown in Figure 34-57 and described in Table 34-52.

Return to the Summary Table.

MCAN Extended ID and Mask

Figure 34-57. MCAN_XIDAM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED			EIDM												
R-0h			R/WQ-1FFFFFFFh												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIDM															
R/WQ-1FFFFFFFh															

Table 34-52. MCAN_XIDAM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	Reserved
28-0	EIDM	R/WQ	1FFFFFFFh	Extended ID Mask. For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn

34.7.3.23 MCAN_HPMS Register (Offset = 94h) [Reset = 00000000h]

MCAN_HPMS is shown in Figure 34-58 and described in Table 34-53.

Return to the Summary Table.

This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.

Figure 34-58. MCAN_HPMS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
FLST	FIDX						
R-0h	R-0h						
7	6	5	4	3	2	1	0
MSI		BIDX					
R-0h		R-0h					

Table 34-53. MCAN_HPMS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	FLST	R	0h	Filter List. Indicates the filter list of the matching filter element. 0 Standard Filter List 1 Extended Filter List Reset type: SYSRSn
14-8	FIDX	R	0h	Filter Index. Index of matching filter element. Range is 0 to SIDFC.LSS - 1 resp. XIDFC.LSE - 1. Reset type: SYSRSn
7-6	MSI	R	0h	Message Storage Indicator 00 No FIFO selected 01 FIFO message lost 10 Message stored in FIFO 0 11 Message stored in FIFO 1 Reset type: SYSRSn
5-0	BIDX	R	0h	Buffer Index. Index of Rx FIFO element to which the message was stored. Only valid when MSI[1] = '1'. Reset type: SYSRSn

34.7.3.24 MCAN_NDAT1 Register (Offset = 98h) [Reset = 00000000h]

MCAN_NDAT1 is shown in Figure 34-59 and described in Table 34-54.

Return to the Summary Table.

MCAN New Data 1

Figure 34-59. MCAN_NDAT1 Register

31	30	29	28	27	26	25	24
ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	16
ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
15	14	13	12	11	10	9	8
ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

Table 34-54. MCAN_NDAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ND31	R/W1C	0h	New Data RX Buffer 31 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
30	ND30	R/W1C	0h	New Data RX Buffer 30 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
29	ND29	R/W1C	0h	New Data RX Buffer 29 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
28	ND28	R/W1C	0h	New Data RX Buffer 28 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
27	ND27	R/W1C	0h	New Data RX Buffer 27 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
26	ND26	R/W1C	0h	New Data RX Buffer 26 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
25	ND25	R/W1C	0h	New Data RX Buffer 25 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn

Table 34-54. MCAN_NDAT1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	ND24	R/W1C	0h	New Data RX Buffer 24 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
23	ND23	R/W1C	0h	New Data RX Buffer 23 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
22	ND22	R/W1C	0h	New Data RX Buffer 22 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
21	ND21	R/W1C	0h	New Data RX Buffer 21 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
20	ND20	R/W1C	0h	New Data RX Buffer 20 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
19	ND19	R/W1C	0h	New Data RX Buffer 19 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
18	ND18	R/W1C	0h	New Data RX Buffer 18 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
17	ND17	R/W1C	0h	New Data RX Buffer 17 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
16	ND16	R/W1C	0h	New Data RX Buffer 16 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
15	ND15	R/W1C	0h	New Data RX Buffer 15 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
14	ND14	R/W1C	0h	New Data RX Buffer 14 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
13	ND13	R/W1C	0h	New Data RX Buffer 13 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
12	ND12	R/W1C	0h	New Data RX Buffer 12 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
11	ND11	R/W1C	0h	New Data RX Buffer 11 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn

Table 34-54. MCAN_NDAT1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	ND10	R/W1C	0h	New Data RX Buffer 10 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
9	ND9	R/W1C	0h	New Data RX Buffer 9 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
8	ND8	R/W1C	0h	New Data RX Buffer 8 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
7	ND7	R/W1C	0h	New Data RX Buffer 7 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
6	ND6	R/W1C	0h	New Data RX Buffer 6 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
5	ND5	R/W1C	0h	New Data RX Buffer 5 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
4	ND4	R/W1C	0h	New Data RX Buffer 4 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
3	ND3	R/W1C	0h	New Data RX Buffer 3 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
2	ND2	R/W1C	0h	New Data RX Buffer 2 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
1	ND1	R/W1C	0h	New Data RX Buffer 1 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
0	ND0	R/W1C	0h	New Data RX Buffer 0 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn

34.7.3.25 MCAN_NDAT2 Register (Offset = 9Ch) [Reset = 00000000h]

MCAN_NDAT2 is shown in Figure 34-60 and described in Table 34-55.

Return to the Summary Table.

MCAN New Data 2

Figure 34-60. MCAN_NDAT2 Register

31	30	29	28	27	26	25	24
ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	16
ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
15	14	13	12	11	10	9	8
ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

Table 34-55. MCAN_NDAT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ND63	R/W1C	0h	New Data RX Buffer 63 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
30	ND62	R/W1C	0h	New Data RX Buffer 62 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
29	ND61	R/W1C	0h	New Data RX Buffer 61 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
28	ND60	R/W1C	0h	New Data RX Buffer 60 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
27	ND59	R/W1C	0h	New Data RX Buffer 59 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
26	ND58	R/W1C	0h	New Data RX Buffer 58 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
25	ND57	R/W1C	0h	New Data RX Buffer 57 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn

Table 34-55. MCAN_NDAT2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	ND56	R/W1C	0h	New Data RX Buffer 56 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
23	ND55	R/W1C	0h	New Data RX Buffer 55 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
22	ND54	R/W1C	0h	New Data RX Buffer 54 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
21	ND53	R/W1C	0h	New Data RX Buffer 53 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
20	ND52	R/W1C	0h	New Data RX Buffer 52 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
19	ND51	R/W1C	0h	New Data RX Buffer 51 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
18	ND50	R/W1C	0h	New Data RX Buffer 50 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
17	ND49	R/W1C	0h	New Data RX Buffer 49 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
16	ND48	R/W1C	0h	New Data RX Buffer 48 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
15	ND47	R/W1C	0h	New Data RX Buffer 47 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
14	ND46	R/W1C	0h	New Data RX Buffer 46 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
13	ND45	R/W1C	0h	New Data RX Buffer 45 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
12	ND44	R/W1C	0h	New Data RX Buffer 44 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
11	ND43	R/W1C	0h	New Data RX Buffer 43 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn

Table 34-55. MCAN_NDAT2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	ND42	R/W1C	0h	New Data RX Buffer 42 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
9	ND41	R/W1C	0h	New Data RX Buffer 41 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
8	ND40	R/W1C	0h	New Data RX Buffer 40 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
7	ND39	R/W1C	0h	New Data RX Buffer 39 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
6	ND38	R/W1C	0h	New Data RX Buffer 38 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
5	ND37	R/W1C	0h	New Data RX Buffer 37 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
4	ND36	R/W1C	0h	New Data RX Buffer 36 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
3	ND35	R/W1C	0h	New Data RX Buffer 35 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
2	ND34	R/W1C	0h	New Data RX Buffer 34 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
1	ND33	R/W1C	0h	New Data RX Buffer 33 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn
0	ND32	R/W1C	0h	New Data RX Buffer 32 0 Rx Buffer not updated 1 Rx Buffer updated from new message Reset type: SYSRSn

34.7.3.26 MCAN_RXF0C Register (Offset = A0h) [Reset = 00000000h]

MCAN_RXF0C is shown in Figure 34-61 and described in Table 34-56.

Return to the Summary Table.

MCAN Rx FIFO 0 Configuration

Figure 34-61. MCAN_RXF0C Register

31	30	29	28	27	26	25	24
F0OM	F0WM						
R/WQ-0h	R/WQ-0h						
23	22	21	20	19	18	17	16
RESERVED	F0S						
R-0h	R/WQ-0h						
15	14	13	12	11	10	9	8
F0SA							
R/WQ-0h							
7	6	5	4	3	2	1	0
F0SA						RESERVED	
R/WQ-0h						R-0h	

Table 34-56. MCAN_RXF0C Register Field Descriptions

Bit	Field	Type	Reset	Description
31	F0OM	R/WQ	0h	FIFO 0 Operation Mode. FIFO 0 can be operated in blocking or in overwrite mode. 0 FIFO 0 blocking mode 1 FIFO 0 overwrite mode Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
30-24	F0WM	R/WQ	0h	Rx FIFO 0 Watermark 0 Watermark interrupt disabled 1-64 Level for Rx FIFO 0 watermark interrupt (IR.RF0W) >64 Watermark interrupt disabled Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
23	RESERVED	R	0h	Reserved
22-16	F0S	R/WQ	0h	Rx FIFO 0 Size. The Rx FIFO 0 elements are indexed from 0 to F0S-1. 0 No Rx FIFO 0 1-64 Number of Rx FIFO 0 elements >64 Values greater than 64 are interpreted as 64 Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
15-2	F0SA	R/WQ	0h	Rx FIFO 0 Start Address. Start address of Rx FIFO 0 in Message RAM (32-bit word address). Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
1-0	RESERVED	R	0h	Reserved

34.7.3.27 MCAN_RXF0S Register (Offset = A4h) [Reset = 00000000h]

MCAN_RXF0S is shown in Figure 34-62 and described in Table 34-57.

Return to the Summary Table.

MCAN Rx FIFO 0 Status

Figure 34-62. MCAN_RXF0S Register

31	30	29	28	27	26	25	24
RESERVED						RF0L	F0F
R-0h						R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED		F0PI					
R-0h		R-0h					
15	14	13	12	11	10	9	8
RESERVED		F0GI					
R-0h		R-0h					
7	6	5	4	3	2	1	0
RESERVED	F0FL						
R-0h	R-0h						

Table 34-57. MCAN_RXF0S Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	RF0L	R	0h	Rx FIFO 0 Message Lost. This bit is a copy of interrupt flag IR.RF0L. When IR.RF0L is reset, this bit is also reset. 0 No Rx FIFO 0 message lost 1 Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero Note: Overwriting the oldest message when RXF0C.F0OM = '1' will not set this flag. Reset type: SYSRSn
24	F0F	R	0h	Rx FIFO 0 Full 0 Rx FIFO 0 not full 1 Rx FIFO 0 full Reset type: SYSRSn
23-22	RESERVED	R	0h	Reserved
21-16	F0PI	R	0h	Rx FIFO 0 Put Index. Rx FIFO 0 write index pointer, range 0 to 63. Reset type: SYSRSn
15-14	RESERVED	R	0h	Reserved
13-8	F0GI	R	0h	Rx FIFO 0 Get Index. Rx FIFO 0 read index pointer, range 0 to 63. Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6-0	F0FL	R	0h	Rx FIFO 0 Fill Level. Number of elements stored in Rx FIFO 0, range 0 to 64. Reset type: SYSRSn

34.7.3.28 MCAN_RXF0A Register (Offset = A8h) [Reset = 00000000h]

MCAN_RXF0A is shown in Figure 34-63 and described in Table 34-58.

Return to the Summary Table.

MCAN Rx FIFO 0 Acknowledge

Figure 34-63. MCAN_RXF0A Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										F0AI					
R-0h																										R/W-0h					

Table 34-58. MCAN_RXF0A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-0	F0AI	R/W	0h	Rx FIFO 0 Acknowledge Index. After the Host has read a message or a sequence of messages from Rx FIFO 0 it has to write the buffer index of the last element read from Rx FIFO 0 to F0AI. This will set the Rx FIFO 0 Get Index RXF0S.F0GI to F0AI + 1 and update the FIFO 0 Fill Level RXF0S.F0FL. Reset type: SYSRSn

34.7.3.29 MCAN_RXBC Register (Offset = ACh) [Reset = 00000000h]

MCAN_RXBC is shown in Figure 34-64 and described in Table 34-59.

Return to the Summary Table.

MCAN Rx Buffer Configuration

Figure 34-64. MCAN_RXBC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RBSA							
R/WQ-0h							
7	6	5	4	3	2	1	0
RBSA						RESERVED	
R/WQ-0h						R-0h	

Table 34-59. MCAN_RXBC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-2	RBSA	R/WQ	0h	Rx Buffer Start Address. Configures the start address of the Rx Buffers section in the Message RAM (32-bit word address). +I466 Reset type: SYSRSn
1-0	RESERVED	R	0h	Reserved

34.7.3.30 MCAN_RXF1C Register (Offset = B0h) [Reset = 00000000h]

MCAN_RXF1C is shown in Figure 34-65 and described in Table 34-60.

Return to the Summary Table.

MCAN Rx FIFO 1 Configuration

Figure 34-65. MCAN_RXF1C Register

31	30	29	28	27	26	25	24
F1OM	F1WM						
R/WQ-0h	R/WQ-0h						
23	22	21	20	19	18	17	16
RESERVED	F1S						
R-0h	R/WQ-0h						
15	14	13	12	11	10	9	8
F1SA							
R/WQ-0h							
7	6	5	4	3	2	1	0
F1SA						RESERVED	
R/WQ-0h						R-0h	

Table 34-60. MCAN_RXF1C Register Field Descriptions

Bit	Field	Type	Reset	Description
31	F1OM	R/WQ	0h	FIFO 1 Operation Mode. FIFO 1 can be operated in blocking or in overwrite mode. 0 FIFO 1 blocking mode 1 FIFO 1 overwrite mode Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
30-24	F1WM	R/WQ	0h	Rx FIFO 1 Watermark 0 Watermark interrupt disabled 1-64 Level for Rx FIFO 1 watermark interrupt (IR.RF1W) >64 Watermark interrupt disabled Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
23	RESERVED	R	0h	Reserved
22-16	F1S	R/WQ	0h	Rx FIFO 1 Size. The Rx FIFO 1 elements are indexed from 0 to F1S - 1. 0 No Rx FIFO 1 1-64 Number of Rx FIFO 1 elements >64 Values greater than 64 are interpreted as 64 Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
15-2	F1SA	R/WQ	0h	Rx FIFO 1 Start Address Start address of Rx FIFO 1 in Message RAM (32-bit word address). Reset type: SYSRSn
1-0	RESERVED	R	0h	Reserved

34.7.3.31 MCAN_RXF1S Register (Offset = B4h) [Reset = 00000000h]

MCAN_RXF1S is shown in Figure 34-66 and described in Table 34-61.

Return to the Summary Table.

MCAN Rx FIFO 1 Status

Figure 34-66. MCAN_RXF1S Register

31	30	29	28	27	26	25	24
DMS		RESERVED				RF1L	F1F
R-0h		R-0h				R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED		F1PI					
R-0h		R-0h					
15	14	13	12	11	10	9	8
RESERVED		F1GI					
R-0h		R-0h					
7	6	5	4	3	2	1	0
RESERVED	F1FL						
R-0h	R-0h						

Table 34-61. MCAN_RXF1S Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	DMS	R	0h	Debug Message Status 00 Idle state, wait for reception of debug messages, DMA request is cleared 01 Debug message A received 10 Debug messages A, B received 11 Debug messages A, B, C received, DMA request is set Reset type: SYSRSn
29-26	RESERVED	R	0h	Reserved
25	RF1L	R	0h	Rx FIFO 1 Message Lost. This bit is a copy of interrupt flag IR.RF1L. When IR.RF1L is reset, this bit is also reset. 0 No Rx FIFO 1 message lost 1 Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero Note: Overwriting the oldest message when RXF1C.F1OM = '1' will not set this flag. Reset type: SYSRSn
24	F1F	R	0h	Rx FIFO 1 Full 0 Rx FIFO 1 not full 1 Rx FIFO 1 full Reset type: SYSRSn
23-22	RESERVED	R	0h	Reserved
21-16	F1PI	R	0h	Rx FIFO 1 Put Index. Rx FIFO 1 write index pointer, range 0 to 63. Reset type: SYSRSn
15-14	RESERVED	R	0h	Reserved
13-8	F1GI	R	0h	Rx FIFO 1 Get Index. Rx FIFO 1 read index pointer, range 0 to 63. Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6-0	F1FL	R	0h	Rx FIFO 1 Fill Level. Number of elements stored in Rx FIFO 1, range 0 to 64. Reset type: SYSRSn

34.7.3.32 MCAN_RXF1A Register (Offset = B8h) [Reset = 00000000h]

MCAN_RXF1A is shown in Figure 34-67 and described in Table 34-62.

Return to the Summary Table.

MCAN Rx FIFO 1 Acknowledge

Figure 34-67. MCAN_RXF1A Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										F1AI					
R-0h																										R/W-0h					

Table 34-62. MCAN_RXF1A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-0	F1AI	R/W	0h	Rx FIFO 1 Acknowledge Index. After the Host has read a message or a sequence of messages from Rx FIFO 1 it has to write the buffer index of the last element read from Rx FIFO 1 to F1AI. This will set the Rx FIFO 1 Get Index RXF1S.F1GI to F1AI + 1 and update the FIFO 1 Fill Level RXF1S.F1FL. Reset type: SYSRSn

34.7.3.33 MCAN_RXESC Register (Offset = BCh) [Reset = 00000000h]

MCAN_RXESC is shown in Figure 34-68 and described in Table 34-63.

Return to the Summary Table.

Configures the number of data bytes belonging to an Rx Buffer / Rx FIFO element. Data field sizes >8 bytes are intended for CAN FD operation only.

Figure 34-68. MCAN_RXESC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED					RBDS		
R-0h					R/WQ-0h		
7	6	5	4	3	2	1	0
RESERVED	F1DS			RESERVED	F0DS		
R-0h	R/WQ-0h			R-0h	R/WQ-0h		

Table 34-63. MCAN_RXESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-11	RESERVED	R	0h	Reserved
10-8	RBDS	R/WQ	0h	Rx Buffer Data Field Size 000 8 byte data field 001 12 byte data field 010 16 byte data field 011 20 byte data field 100 24 byte data field 101 32 byte data field 110 48 byte data field 111 64 byte data field Note: In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by RXESC are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame's data field is ignored. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
7	RESERVED	R	0h	Reserved

Table 34-63. MCAN_RXESC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6-4	F1DS	R/WQ	0h	Rx FIFO 1 Data Field Size 000 8 byte data field 001 12 byte data field 010 16 byte data field 011 20 byte data field 100 24 byte data field 101 32 byte data field 110 48 byte data field 111 64 byte data field Note: In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by RXESC are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame's data field is ignored. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
3	RESERVED	R	0h	Reserved
2-0	F0DS	R/WQ	0h	Rx FIFO 0 Data Field Size 000 8 byte data field 001 12 byte data field 010 16 byte data field 011 20 byte data field 100 24 byte data field 101 32 byte data field 110 48 byte data field 111 64 byte data field Note: In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by RXESC are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame's data field is ignored. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn

34.7.3.34 MCAN_TXBC Register (Offset = C0h) [Reset = 00000000h]

MCAN_TXBC is shown in Figure 34-69 and described in Table 34-64.

Return to the Summary Table.

MCAN Tx Buffer Configuration

Figure 34-69. MCAN_TXBC Register

31	30	29	28	27	26	25	24
RESERVED	TFQM	TFQS					
R-0h	R/WQ-0h	R/WQ-0h					
23	22	21	20	19	18	17	16
RESERVED		NDTB					
R-0h		R/WQ-0h					
15	14	13	12	11	10	9	8
TBSA							
R/WQ-0h							
7	6	5	4	3	2	1	0
TBSA						RESERVED	
R/WQ-0h						R-0h	

Table 34-64. MCAN_TXBC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	TFQM	R/WQ	0h	Tx FIFO/Queue Mode 0 Tx FIFO operation 1 Tx Queue operation Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
29-24	TFQS	R/WQ	0h	Transmit FIFO/Queue Size 0 No Tx FIFO/Queue 1-32 Number of Tx Buffers used for Tx FIFO/Queue >32 Values greater than 32 are interpreted as 32 Note: Be aware that the sum of TFQS and NDTB may be not greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
23-22	RESERVED	R	0h	Reserved
21-16	NDTB	R/WQ	0h	Number of Dedicated Transmit Buffers 0 No Dedicated Tx Buffers 1-32 Number of Dedicated Tx Buffers >32 Values greater than 32 are interpreted as 32 Note: Be aware that the sum of TFQS and NDTB may be not greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn

Table 34-64. MCAN_TXBC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-2	TBSA	R/WQ	0h	Tx Buffers Start Address. Start address of Tx Buffers section in Message RAM (32-bit word address). Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn
1-0	RESERVED	R	0h	Reserved

34.7.3.35 MCAN_TXFQS Register (Offset = C4h) [Reset = 00000000h]

MCAN_TXFQS is shown in Figure 34-70 and described in Table 34-65.

Return to the Summary Table.

The Tx FIFO/Queue status is related to the pending Tx requests listed in register TXBRP. Therefore the effect of Add/Cancellation requests may be delayed due to a running Tx scan (TXBRP not yet updated).

Figure 34-70. MCAN_TXFQS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED		TFQF	TFQP				
R-0h		R-0h	R-0h				
15	14	13	12	11	10	9	8
RESERVED			TFGI				
R-0h			R-0h				
7	6	5	4	3	2	1	0
RESERVED		TFFL					
R-0h		R-0h					

Table 34-65. MCAN_TXFQS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	Reserved
21	TFQF	R	0h	Tx FIFO/Queue Full 0 Tx FIFO/Queue not full 1 Tx FIFO/Queue full Reset type: SYSRSn
20-16	TFQP	R	0h	Tx FIFO/Queue Put Index. Tx FIFO/Queue write index pointer, range 0 to 31. Note: In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers. Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO. Reset type: SYSRSn
15-13	RESERVED	R	0h	Reserved
12-8	TFGI	R	0h	Tx FIFO Get Index. Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (TXBC.TFQM = '1'). Note: In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers. Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO. Reset type: SYSRSn
7-6	RESERVED	R	0h	Reserved
5-0	TFFL	R	0h	Tx FIFO Free Level. Number of consecutive free Tx FIFO elements starting from TFGI, range 0 to 32. Read as zero when Tx Queue operation is configured (TXBC.TFQM = '1'). Reset type: SYSRSn

34.7.3.36 MCAN_TXESC Register (Offset = C8h) [Reset = 00000000h]

MCAN_TXESC is shown in Figure 34-71 and described in Table 34-66.

Return to the Summary Table.

Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes > 8 bytes are intended for CAN FD operation only.

Figure 34-71. MCAN_TXESC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													TBDS		
R-0h													R/WQ-0h		

Table 34-66. MCAN_TXESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	TBDS	R/WQ	0h	Tx Buffer Data Field Size 000 8 byte data field 001 12 byte data field 010 16 byte data field 011 20 byte data field 100 24 byte data field 101 32 byte data field 110 48 byte data field 111 64 byte data field Note: In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size TXESC.TBDS, the bytes not defined by the Tx Buffer are transmitted as '0xCC' (padding bytes). Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. Reset type: SYSRSn

34.7.3.37 MCAN_TXBRP Register (Offset = CCh) [Reset = 00000000h]

MCAN_TXBRP is shown in Figure 34-72 and described in Table 34-67.

Return to the Summary Table.

MCAN Tx Buffer Request Pending

Figure 34-72. MCAN_TXBRP Register

31	30	29	28	27	26	25	24
TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 34-67. MCAN_TXBRP Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TRP31	R	0h	Transmission Request Pending 31. See description for bit 0. Reset type: SYSRSn
30	TRP30	R	0h	Transmission Request Pending 30. See description for bit 0. Reset type: SYSRSn
29	TRP29	R	0h	Transmission Request Pending 29. See description for bit 0. Reset type: SYSRSn
28	TRP28	R	0h	Transmission Request Pending 28. See description for bit 0. Reset type: SYSRSn
27	TRP27	R	0h	Transmission Request Pending 27. See description for bit 0. Reset type: SYSRSn
26	TRP26	R	0h	Transmission Request Pending 26. See description for bit 0. Reset type: SYSRSn
25	TRP25	R	0h	Transmission Request Pending 25. See description for bit 0. Reset type: SYSRSn
24	TRP24	R	0h	Transmission Request Pending 24. See description for bit 0. Reset type: SYSRSn
23	TRP23	R	0h	Transmission Request Pending 23. See description for bit 0. Reset type: SYSRSn
22	TRP22	R	0h	Transmission Request Pending 22. See description for bit 0. Reset type: SYSRSn
21	TRP21	R	0h	Transmission Request Pending 21. See description for bit 0. Reset type: SYSRSn
20	TRP20	R	0h	Transmission Request Pending 20. See description for bit 0. Reset type: SYSRSn
19	TRP19	R	0h	Transmission Request Pending 19. See description for bit 0. Reset type: SYSRSn

Table 34-67. MCAN_TXBRP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	TRP18	R	0h	Transmission Request Pending 18. See description for bit 0. Reset type: SYSRSn
17	TRP17	R	0h	Transmission Request Pending 17. See description for bit 0. Reset type: SYSRSn
16	TRP16	R	0h	Transmission Request Pending 16. See description for bit 0. Reset type: SYSRSn
15	TRP15	R	0h	Transmission Request Pending 15. See description for bit 0. Reset type: SYSRSn
14	TRP14	R	0h	Transmission Request Pending 14. See description for bit 0. Reset type: SYSRSn
13	TRP13	R	0h	Transmission Request Pending 13. See description for bit 0. Reset type: SYSRSn
12	TRP12	R	0h	Transmission Request Pending 12. See description for bit 0. Reset type: SYSRSn
11	TRP11	R	0h	Transmission Request Pending 11. See description for bit 0. Reset type: SYSRSn
10	TRP10	R	0h	Transmission Request Pending 10. See description for bit 0. Reset type: SYSRSn
9	TRP9	R	0h	Transmission Request Pending 9. See description for bit 0. Reset type: SYSRSn
8	TRP8	R	0h	Transmission Request Pending 8. See description for bit 0. Reset type: SYSRSn
7	TRP7	R	0h	Transmission Request Pending 7. See description for bit 0. Reset type: SYSRSn
6	TRP6	R	0h	Transmission Request Pending 6. See description for bit 0. Reset type: SYSRSn
5	TRP5	R	0h	Transmission Request Pending 5. See description for bit 0. Reset type: SYSRSn
4	TRP4	R	0h	Transmission Request Pending 4. See description for bit 0. Reset type: SYSRSn
3	TRP3	R	0h	Transmission Request Pending 3. See description for bit 0. Reset type: SYSRSn
2	TRP2	R	0h	Transmission Request Pending 2. See description for bit 0. Reset type: SYSRSn
1	TRP1	R	0h	Transmission Request Pending 1. See description for bit 0. Reset type: SYSRSn

Table 34-67. MCAN_TXBRP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	TRP0	R	0h	Transmission Request Pending 0. Each Tx Buffer has its own Transmission Request Pending bit. The bits are set via register TXBAR. The bits are reset after a requested transmission has completed or has been cancelled via register TXBCR. TXBRP bits are set only for those Tx Buffers configured via TXBC. After a TXBRP bit has been set, a Tx scan is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID). A cancellation request resets the corresponding transmission request pending bit of register TXBRP. In case a transmission has already been started when a cancellation is requested, this is done at the end of the transmission, regardless whether the transmission was successful or not. The cancellation request bits are reset directly after the corresponding TXBRP bit has been reset. After a cancellation has been requested, a finished cancellation is signalled via TXBCF - after successful transmission together with the corresponding TXBTO bit - when the transmission has not yet been started at the point of cancellation - when the transmission has been aborted due to lost arbitration - when an error occurred during frame transmission In DAR mode all transmissions are automatically cancelled if they are not successful. The corresponding TXBCF bit is set for all unsuccessful transmissions. 0 No transmission request pending 1 Transmission request pending Note: TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is cancelled immediately, the corresponding TXBRP bit is reset. Reset type: SYSRSn

34.7.3.38 MCAN_TXBAR Register (Offset = D0h) [Reset = 00000000h]

MCAN_TXBAR is shown in Figure 34-73 and described in Table 34-68.

Return to the Summary Table.

MCAN Tx Buffer Add Request

Figure 34-73. MCAN_TXBAR Register

31	30	29	28	27	26	25	24
AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h
23	22	21	20	19	18	17	16
AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h
15	14	13	12	11	10	9	8
AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h
7	6	5	4	3	2	1	0
AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h

Table 34-68. MCAN_TXBAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AR31	R/WQ	0h	Add Request 31. See description for bit 0. Reset type: SYSRSn
30	AR30	R/WQ	0h	Add Request 30. See description for bit 0. Reset type: SYSRSn
29	AR29	R/WQ	0h	Add Request 29. See description for bit 0. Reset type: SYSRSn
28	AR28	R/WQ	0h	Add Request 28. See description for bit 0. Reset type: SYSRSn
27	AR27	R/WQ	0h	Add Request 27. See description for bit 0. Reset type: SYSRSn
26	AR26	R/WQ	0h	Add Request 26. See description for bit 0. Reset type: SYSRSn
25	AR25	R/WQ	0h	Add Request 25. See description for bit 0. Reset type: SYSRSn
24	AR24	R/WQ	0h	Add Request 24. See description for bit 0. Reset type: SYSRSn
23	AR23	R/WQ	0h	Add Request 23. See description for bit 0. Reset type: SYSRSn
22	AR22	R/WQ	0h	Add Request 22. See description for bit 0. Reset type: SYSRSn
21	AR21	R/WQ	0h	Add Request 21. See description for bit 0. Reset type: SYSRSn
20	AR20	R/WQ	0h	Add Request 20. See description for bit 0. Reset type: SYSRSn
19	AR19	R/WQ	0h	Add Request 19. See description for bit 0. Reset type: SYSRSn

Table 34-68. MCAN_TXBAR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	AR18	R/WQ	0h	Add Request 18. See description for bit 0. Reset type: SYSRSn
17	AR17	R/WQ	0h	Add Request 17. See description for bit 0. Reset type: SYSRSn
16	AR16	R/WQ	0h	Add Request 16. See description for bit 0. Reset type: SYSRSn
15	AR15	R/WQ	0h	Add Request 15. See description for bit 0. Reset type: SYSRSn
14	AR14	R/WQ	0h	Add Request 14. See description for bit 0. Reset type: SYSRSn
13	AR13	R/WQ	0h	Add Request 13. See description for bit 0. Reset type: SYSRSn
12	AR12	R/WQ	0h	Add Request 12. See description for bit 0. Reset type: SYSRSn
11	AR11	R/WQ	0h	Add Request 11. See description for bit 0. Reset type: SYSRSn
10	AR10	R/WQ	0h	Add Request 10. See description for bit 0. Reset type: SYSRSn
9	AR9	R/WQ	0h	Add Request 9. See description for bit 0. Reset type: SYSRSn
8	AR8	R/WQ	0h	Add Request 8. See description for bit 0. Reset type: SYSRSn
7	AR7	R/WQ	0h	Add Request 7. See description for bit 0. Reset type: SYSRSn
6	AR6	R/WQ	0h	Add Request 6. See description for bit 0. Reset type: SYSRSn
5	AR5	R/WQ	0h	Add Request 5. See description for bit 0. Reset type: SYSRSn
4	AR4	R/WQ	0h	Add Request 4. See description for bit 0. Reset type: SYSRSn
3	AR3	R/WQ	0h	Add Request 3. See description for bit 0. Reset type: SYSRSn
2	AR2	R/WQ	0h	Add Request 2. See description for bit 0. Reset type: SYSRSn
1	AR1	R/WQ	0h	Add Request 1. See description for bit 0. Reset type: SYSRSn
0	AR0	R/WQ	0h	Add Request 0. Each Tx Buffer has its own Add Request bit. Writing a '1' will set the corresponding Add Request bit writing a '0' has no impact. This enables the Host to set transmission requests for multiple Tx Buffers with one write to TXBAR. TXBAR bits are set only for those Tx Buffers configured via TXBC. When no Tx scan is running, the bits are reset immediately, else the bits remain set until the Tx scan process has completed. 0 No transmission request added 1 Transmission requested added Note: If an add request is applied for a Tx Buffer with pending transmission request (corresponding TXBRP bit already set), this add request is ignored. Qualified Write is possible only with CCCR.CCE='0' Reset type: SYSRSn

34.7.3.39 MCAN_TXBCR Register (Offset = D4h) [Reset = 00000000h]

MCAN_TXBCR is shown in Figure 34-74 and described in Table 34-69.

Return to the Summary Table.

MCAN Tx Buffer Cancellation Request

Figure 34-74. MCAN_TXBCR Register

31	30	29	28	27	26	25	24
CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h
23	22	21	20	19	18	17	16
CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h
15	14	13	12	11	10	9	8
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h
7	6	5	4	3	2	1	0
CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h

Table 34-69. MCAN_TXBCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CR31	R/WQ	0h	Cancellation Request 31. See description for bit 0. Reset type: SYSRSn
30	CR30	R/WQ	0h	Cancellation Request 30. See description for bit 0. Reset type: SYSRSn
29	CR29	R/WQ	0h	Cancellation Request 29. See description for bit 0. Reset type: SYSRSn
28	CR28	R/WQ	0h	Cancellation Request 28. See description for bit 0. Reset type: SYSRSn
27	CR27	R/WQ	0h	Cancellation Request 27. See description for bit 0. Reset type: SYSRSn
26	CR26	R/WQ	0h	Cancellation Request 26. See description for bit 0. Reset type: SYSRSn
25	CR25	R/WQ	0h	Cancellation Request 25. See description for bit 0. Reset type: SYSRSn
24	CR24	R/WQ	0h	Cancellation Request 24. See description for bit 0. Reset type: SYSRSn
23	CR23	R/WQ	0h	Cancellation Request 23. See description for bit 0. Reset type: SYSRSn
22	CR22	R/WQ	0h	Cancellation Request 22. See description for bit 0. Reset type: SYSRSn
21	CR21	R/WQ	0h	Cancellation Request 21. See description for bit 0. Reset type: SYSRSn
20	CR20	R/WQ	0h	Cancellation Request 20. See description for bit 0. Reset type: SYSRSn
19	CR19	R/WQ	0h	Cancellation Request 19. See description for bit 0. Reset type: SYSRSn

Table 34-69. MCAN_TXBCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	CR18	R/WQ	0h	Cancellation Request 18. See description for bit 0. Reset type: SYSRSn
17	CR17	R/WQ	0h	Cancellation Request 17. See description for bit 0. Reset type: SYSRSn
16	CR16	R/WQ	0h	Cancellation Request 16. See description for bit 0. Reset type: SYSRSn
15	CR15	R/WQ	0h	Cancellation Request 15. See description for bit 0. Reset type: SYSRSn
14	CR14	R/WQ	0h	Cancellation Request 14. See description for bit 0. Reset type: SYSRSn
13	CR13	R/WQ	0h	Cancellation Request 13. See description for bit 0. Reset type: SYSRSn
12	CR12	R/WQ	0h	Cancellation Request 12. See description for bit 0. Reset type: SYSRSn
11	CR11	R/WQ	0h	Cancellation Request 11. See description for bit 0. Reset type: SYSRSn
10	CR10	R/WQ	0h	Cancellation Request 10. See description for bit 0. Reset type: SYSRSn
9	CR9	R/WQ	0h	Cancellation Request 9. See description for bit 0. Reset type: SYSRSn
8	CR8	R/WQ	0h	Cancellation Request 8. See description for bit 0. Reset type: SYSRSn
7	CR7	R/WQ	0h	Cancellation Request 7. See description for bit 0. Reset type: SYSRSn
6	CR6	R/WQ	0h	Cancellation Request 6. See description for bit 0. Reset type: SYSRSn
5	CR5	R/WQ	0h	Cancellation Request 5. See description for bit 0. Reset type: SYSRSn
4	CR4	R/WQ	0h	Cancellation Request 4. See description for bit 0. Reset type: SYSRSn
3	CR3	R/WQ	0h	Cancellation Request 3. See description for bit 0. Reset type: SYSRSn
2	CR2	R/WQ	0h	Cancellation Request 2. See description for bit 0. Reset type: SYSRSn
1	CR1	R/WQ	0h	Cancellation Request 1. See description for bit 0. Reset type: SYSRSn
0	CR0	R/WQ	0h	Cancellation Request 0. Each Tx Buffer has its own Cancellation Request bit. Writing a '1' will set the corresponding Cancellation Request bit writing a '0' has no impact. This enables the Host to set cancellation requests for multiple Tx Buffers with one write to TXBCR. TXBCR bits are set only for those Tx Buffers configured via TXBC. The bits remain set until the corresponding bit of TXBRP is reset. 0 No cancellation pending 1 Cancellation pending Qualified Write is possible only with CCCR.CCE='0' Reset type: SYSRSn

34.7.3.40 MCAN_TXBTO Register (Offset = D8h) [Reset = 00000000h]

MCAN_TXBTO is shown in Figure 34-75 and described in Table 34-70.

Return to the Summary Table.

MCAN Tx Buffer Transmission Occurred

Figure 34-75. MCAN_TXBTO Register

31	30	29	28	27	26	25	24
TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 34-70. MCAN_TXBTO Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TO31	R	0h	Transmission Occurred 31. See description for bit 0. Reset type: SYSRSn
30	TO30	R	0h	Transmission Occurred 30. See description for bit 0. Reset type: SYSRSn
29	TO29	R	0h	Transmission Occurred 29. See description for bit 0. Reset type: SYSRSn
28	TO28	R	0h	Transmission Occurred 28. See description for bit 0. Reset type: SYSRSn
27	TO27	R	0h	Transmission Occurred 27. See description for bit 0. Reset type: SYSRSn
26	TO26	R	0h	Transmission Occurred 26. See description for bit 0. Reset type: SYSRSn
25	TO25	R	0h	Transmission Occurred 25. See description for bit 0. Reset type: SYSRSn
24	TO24	R	0h	Transmission Occurred 24. See description for bit 0. Reset type: SYSRSn
23	TO23	R	0h	Transmission Occurred 23. See description for bit 0. Reset type: SYSRSn
22	TO22	R	0h	Transmission Occurred 22. See description for bit 0. Reset type: SYSRSn
21	TO21	R	0h	Transmission Occurred 21. See description for bit 0. Reset type: SYSRSn
20	TO20	R	0h	Transmission Occurred 20. See description for bit 0. Reset type: SYSRSn
19	TO19	R	0h	Transmission Occurred 19. See description for bit 0. Reset type: SYSRSn

Table 34-70. MCAN_TXBTO Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	TO18	R	0h	Transmission Occurred 18. See description for bit 0. Reset type: SYSRSn
17	TO17	R	0h	Transmission Occurred 17. See description for bit 0. Reset type: SYSRSn
16	TO16	R	0h	Transmission Occurred 16. See description for bit 0. Reset type: SYSRSn
15	TO15	R	0h	Transmission Occurred 15. See description for bit 0. Reset type: SYSRSn
14	TO14	R	0h	Transmission Occurred 14. See description for bit 0. Reset type: SYSRSn
13	TO13	R	0h	Transmission Occurred 13. See description for bit 0. Reset type: SYSRSn
12	TO12	R	0h	Transmission Occurred 12. See description for bit 0. Reset type: SYSRSn
11	TO11	R	0h	Transmission Occurred 11. See description for bit 0. Reset type: SYSRSn
10	TO10	R	0h	Transmission Occurred 10. See description for bit 0. Reset type: SYSRSn
9	TO9	R	0h	Transmission Occurred 9. See description for bit 0. Reset type: SYSRSn
8	TO8	R	0h	Transmission Occurred 8. See description for bit 0. Reset type: SYSRSn
7	TO7	R	0h	Transmission Occurred 7. See description for bit 0. Reset type: SYSRSn
6	TO6	R	0h	Transmission Occurred 6. See description for bit 0. Reset type: SYSRSn
5	TO5	R	0h	Transmission Occurred 5. See description for bit 0. Reset type: SYSRSn
4	TO4	R	0h	Transmission Occurred 4. See description for bit 0. Reset type: SYSRSn
3	TO3	R	0h	Transmission Occurred 3. See description for bit 0. Reset type: SYSRSn
2	TO2	R	0h	Transmission Occurred 2. See description for bit 0. Reset type: SYSRSn
1	TO1	R	0h	Transmission Occurred 1. See description for bit 0. Reset type: SYSRSn
0	TO0	R	0h	Transmission Occurred 0. Each Tx Buffer has its own Transmission Occurred bit. The bits are set when the corresponding TXBRP bit is cleared after a successful transmission. The bits are reset when a new transmission is requested by writing a '1' to the corresponding bit of register TXBAR. 0 No transmission occurred 1 Transmission occurred Reset type: SYSRSn

34.7.3.41 MCAN_TXBCF Register (Offset = DCh) [Reset = 00000000h]

MCAN_TXBCF is shown in Figure 34-76 and described in Table 34-71.

Return to the Summary Table.

MCAN Tx Buffer Cancellation Finished

Figure 34-76. MCAN_TXBCF Register

31	30	29	28	27	26	25	24
CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 34-71. MCAN_TXBCF Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CF31	R	0h	Cancellation Finished 31. See description for bit 0. Reset type: SYSRSn
30	CF30	R	0h	Cancellation Finished 30. See description for bit 0. Reset type: SYSRSn
29	CF29	R	0h	Cancellation Finished 29. See description for bit 0. Reset type: SYSRSn
28	CF28	R	0h	Cancellation Finished 28. See description for bit 0. Reset type: SYSRSn
27	CF27	R	0h	Cancellation Finished 27. See description for bit 0. Reset type: SYSRSn
26	CF26	R	0h	Cancellation Finished 26. See description for bit 0. Reset type: SYSRSn
25	CF25	R	0h	Cancellation Finished 25. See description for bit 0. Reset type: SYSRSn
24	CF24	R	0h	Cancellation Finished 24. See description for bit 0. Reset type: SYSRSn
23	CF23	R	0h	Cancellation Finished 23. See description for bit 0. Reset type: SYSRSn
22	CF22	R	0h	Cancellation Finished 22. See description for bit 0. Reset type: SYSRSn
21	CF21	R	0h	Cancellation Finished 21. See description for bit 0. Reset type: SYSRSn
20	CF20	R	0h	Cancellation Finished 20. See description for bit 0. Reset type: SYSRSn
19	CF19	R	0h	Cancellation Finished 19. See description for bit 0. Reset type: SYSRSn

Table 34-71. MCAN_TXBCF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	CF18	R	0h	Cancellation Finished 18. See description for bit 0. Reset type: SYSRSn
17	CF17	R	0h	Cancellation Finished 17. See description for bit 0. Reset type: SYSRSn
16	CF16	R	0h	Cancellation Finished 16. See description for bit 0. Reset type: SYSRSn
15	CF15	R	0h	Cancellation Finished 15. See description for bit 0. Reset type: SYSRSn
14	CF14	R	0h	Cancellation Finished 14. See description for bit 0. Reset type: SYSRSn
13	CF13	R	0h	Cancellation Finished 13. See description for bit 0. Reset type: SYSRSn
12	CF12	R	0h	Cancellation Finished 12. See description for bit 0. Reset type: SYSRSn
11	CF11	R	0h	Cancellation Finished 11. See description for bit 0. Reset type: SYSRSn
10	CF10	R	0h	Cancellation Finished 10. See description for bit 0. Reset type: SYSRSn
9	CF9	R	0h	Cancellation Finished 9. See description for bit 0. Reset type: SYSRSn
8	CF8	R	0h	Cancellation Finished 8. See description for bit 0. Reset type: SYSRSn
7	CF7	R	0h	Cancellation Finished 7. See description for bit 0. Reset type: SYSRSn
6	CF6	R	0h	Cancellation Finished 6. See description for bit 0. Reset type: SYSRSn
5	CF5	R	0h	Cancellation Finished 5. See description for bit 0. Reset type: SYSRSn
4	CF4	R	0h	Cancellation Finished 4. See description for bit 0. Reset type: SYSRSn
3	CF3	R	0h	Cancellation Finished 3. See description for bit 0. Reset type: SYSRSn
2	CF2	R	0h	Cancellation Finished 2. See description for bit 0. Reset type: SYSRSn
1	CF1	R	0h	Cancellation Finished 1. See description for bit 0. Reset type: SYSRSn
0	CF0	R	0h	Cancellation Finished 0. Each Tx Buffer has its own Cancellation Finished bit. The bits are set when the corresponding TXBRP bit is cleared after a cancellation was requested via TXBCR. In case the corresponding TXBRP bit was not set at the point of cancellation, CF is set immediately. The bits are reset when a new transmission is requested by writing a '1' to the corresponding bit of register TXBAR. 0 No transmit buffer cancellation 1 Transmit buffer cancellation finished Reset type: SYSRSn

34.7.3.42 MCAN_TXBTIE Register (Offset = E0h) [Reset = 00000000h]

MCAN_TXBTIE is shown in Figure 34-77 and described in Table 34-72.

Return to the Summary Table.

MCAN Tx Buffer Transmission Interrupt Enable

Figure 34-77. MCAN_TXBTIE Register

31	30	29	28	27	26	25	24
TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 34-72. MCAN_TXBTIE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TIE31	R/W	0h	Transmission Interrupt Enable 31. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
30	TIE30	R/W	0h	Transmission Interrupt Enable 30. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
29	TIE29	R/W	0h	Transmission Interrupt Enable 29. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
28	TIE28	R/W	0h	Transmission Interrupt Enable 28. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
27	TIE27	R/W	0h	Transmission Interrupt Enable 27. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
26	TIE26	R/W	0h	Transmission Interrupt Enable 26. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn

Table 34-72. MCAN_TXBTIE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	TIE25	R/W	0h	Transmission Interrupt Enable 25. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
24	TIE24	R/W	0h	Transmission Interrupt Enable 24. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
23	TIE23	R/W	0h	Transmission Interrupt Enable 23. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
22	TIE22	R/W	0h	Transmission Interrupt Enable 22. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
21	TIE21	R/W	0h	Transmission Interrupt Enable 21. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
20	TIE20	R/W	0h	Transmission Interrupt Enable 20. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
19	TIE19	R/W	0h	Transmission Interrupt Enable 19. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
18	TIE18	R/W	0h	Transmission Interrupt Enable 18. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
17	TIE17	R/W	0h	Transmission Interrupt Enable 17. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
16	TIE16	R/W	0h	Transmission Interrupt Enable 16. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
15	TIE15	R/W	0h	Transmission Interrupt Enable 15. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn

Table 34-72. MCAN_TXBTIE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	TIE14	R/W	0h	Transmission Interrupt Enable 14. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
13	TIE13	R/W	0h	Transmission Interrupt Enable 13. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
12	TIE12	R/W	0h	Transmission Interrupt Enable 12. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
11	TIE11	R/W	0h	Transmission Interrupt Enable 11. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
10	TIE10	R/W	0h	Transmission Interrupt Enable 10. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
9	TIE9	R/W	0h	Transmission Interrupt Enable 9. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
8	TIE8	R/W	0h	Transmission Interrupt Enable 8. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
7	TIE7	R/W	0h	Transmission Interrupt Enable 7. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
6	TIE6	R/W	0h	Transmission Interrupt Enable 6. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
5	TIE5	R/W	0h	Transmission Interrupt Enable 5. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
4	TIE4	R/W	0h	Transmission Interrupt Enable 4. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn

Table 34-72. MCAN_TXBTIE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	TIE3	R/W	0h	Transmission Interrupt Enable 3. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
2	TIE2	R/W	0h	Transmission Interrupt Enable 2. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
1	TIE1	R/W	0h	Transmission Interrupt Enable 1. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn
0	TIE0	R/W	0h	Transmission Interrupt Enable 0. 0 Transmission interrupt disabled 1 Transmission interrupt enable Reset type: SYSRSn

34.7.3.43 MCAN_TXBCIE Register (Offset = E4h) [Reset = 00000000h]

MCAN_TXBCIE is shown in Figure 34-78 and described in Table 34-73.

Return to the Summary Table.

MCAN Tx Buffer Cancellation Finished Interrupt Enable

Figure 34-78. MCAN_TXBCIE Register

31	30	29	28	27	26	25	24
CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 34-73. MCAN_TXBCIE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CFIE31	R/W	0h	Cancellation Finished Interrupt Enable 31. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
30	CFIE30	R/W	0h	Cancellation Finished Interrupt Enable 30. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
29	CFIE29	R/W	0h	Cancellation Finished Interrupt Enable 29. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
28	CFIE28	R/W	0h	Cancellation Finished Interrupt Enable 28. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
27	CFIE27	R/W	0h	Cancellation Finished Interrupt Enable 27. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
26	CFIE26	R/W	0h	Cancellation Finished Interrupt Enable 26. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn

Table 34-73. MCAN_TXBCIE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	CFIE25	R/W	0h	Cancellation Finished Interrupt Enable 25. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
24	CFIE24	R/W	0h	Cancellation Finished Interrupt Enable 24. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
23	CFIE23	R/W	0h	Cancellation Finished Interrupt Enable 23. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
22	CFIE22	R/W	0h	Cancellation Finished Interrupt Enable 22. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
21	CFIE21	R/W	0h	Cancellation Finished Interrupt Enable 21. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
20	CFIE20	R/W	0h	Cancellation Finished Interrupt Enable 20. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
19	CFIE19	R/W	0h	Cancellation Finished Interrupt Enable 19. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
18	CFIE18	R/W	0h	Cancellation Finished Interrupt Enable 18. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
17	CFIE17	R/W	0h	Cancellation Finished Interrupt Enable 17. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
16	CFIE16	R/W	0h	Cancellation Finished Interrupt Enable 16. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
15	CFIE15	R/W	0h	Cancellation Finished Interrupt Enable 15. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn

Table 34-73. MCAN_TXBCIE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	CFIE14	R/W	0h	Cancellation Finished Interrupt Enable 14. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
13	CFIE13	R/W	0h	Cancellation Finished Interrupt Enable 13. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
12	CFIE12	R/W	0h	Cancellation Finished Interrupt Enable 12. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
11	CFIE11	R/W	0h	Cancellation Finished Interrupt Enable 11. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
10	CFIE10	R/W	0h	Cancellation Finished Interrupt Enable 10. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
9	CFIE9	R/W	0h	Cancellation Finished Interrupt Enable 9. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
8	CFIE8	R/W	0h	Cancellation Finished Interrupt Enable 8. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
7	CFIE7	R/W	0h	Cancellation Finished Interrupt Enable 7. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
6	CFIE6	R/W	0h	Cancellation Finished Interrupt Enable 6. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
5	CFIE5	R/W	0h	Cancellation Finished Interrupt Enable 5. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
4	CFIE4	R/W	0h	Cancellation Finished Interrupt Enable 4. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn

Table 34-73. MCAN_TXBCIE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	CFIE3	R/W	0h	Cancellation Finished Interrupt Enable 3. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
2	CFIE2	R/W	0h	Cancellation Finished Interrupt Enable 2. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
1	CFIE1	R/W	0h	Cancellation Finished Interrupt Enable 1. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn
0	CFIE0	R/W	0h	Cancellation Finished Interrupt Enable 0. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled Reset type: SYSRSn

34.7.3.44 MCAN_TXEFC Register (Offset = F0h) [Reset = 00000000h]

MCAN_TXEFC is shown in Figure 34-79 and described in Table 34-74.

Return to the Summary Table.

MCAN Tx Event FIFO Configuration

Figure 34-79. MCAN_TXEFC Register

31	30	29	28	27	26	25	24
RESERVED		EFWM					
R-0h		R/WQ-0h					
23	22	21	20	19	18	17	16
RESERVED		EFS					
R-0h		R/WQ-0h					
15	14	13	12	11	10	9	8
EFSA							
R/WQ-0h							
7	6	5	4	3	2	1	0
EFSA						RESERVED	
R/WQ-0h						R-0h	

Table 34-74. MCAN_TXEFC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Reserved
29-24	EFWM	R/WQ	0h	Event FIFO Watermark 0 Watermark interrupt disabled 1-32 Level for Tx Event FIFO watermark interrupt (IR.TEFW) >32 Watermark interrupt disabled Reset type: SYSRSn
23-22	RESERVED	R	0h	Reserved
21-16	EFS	R/WQ	0h	Event FIFO Size. The Tx Event FIFO elements are indexed from 0 to EFS - 1. 0 Tx Event FIFO disabled 1-32 Number of Tx Event FIFO elements >32 Values greater than 32 are interpreted as 32 Reset type: SYSRSn
15-2	EFSA	R/WQ	0h	Event FIFO Start Address. Start address of Tx Event FIFO in Message RAM (32-bit word address). Reset type: SYSRSn
1-0	RESERVED	R	0h	Reserved

34.7.3.45 MCAN_TXEFS Register (Offset = F4h) [Reset = 00000000h]

MCAN_TXEFS is shown in Figure 34-80 and described in Table 34-75.

Return to the Summary Table.

MCAN Tx Event FIFO Status

Figure 34-80. MCAN_TXEFS Register

31	30	29	28	27	26	25	24
RESERVED						TEFL	EFF
R-0h						R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED			EFPI				
R-0h			R-0h				
15	14	13	12	11	10	9	8
RESERVED			EFGI				
R-0h			R-0h				
7	6	5	4	3	2	1	0
RESERVED		EFFL					
R-0h		R-0h					

Table 34-75. MCAN_TXEFS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	TEFL	R	0h	Tx Event FIFO Element Lost. This bit is a copy of interrupt flag IR.TEFL. When IR.TEFL is reset, this bit is also reset. 0 No Tx Event FIFO element lost 1 Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero. Reset type: SYSRSn
24	EFF	R	0h	Event FIFO Full 0 Tx Event FIFO not full 1 Tx Event FIFO full Reset type: SYSRSn
23-21	RESERVED	R	0h	Reserved
20-16	EFPI	R	0h	Event FIFO Put Index.Tx Event FIFO write index pointer, range 0 to 31. Reset type: SYSRSn
15-13	RESERVED	R	0h	Reserved
12-8	EFGI	R	0h	Event FIFO Get Index. Tx Event FIFO read index pointer, range 0 to 31. Reset type: SYSRSn
7-6	RESERVED	R	0h	Reserved
5-0	EFFL	R	0h	Event FIFO Fill Level. Number of elements stored in Tx Event FIFO, range 0 to 32. Reset type: SYSRSn

34.7.3.46 MCAN_TXEFA Register (Offset = F8h) [Reset = 00000000h]

MCAN_TXEFA is shown in Figure 34-81 and described in Table 34-76.

Return to the Summary Table.

MCAN Tx Event FIFO Acknowledge

Figure 34-81. MCAN_TXEFA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											EFAI				
R-0h																											R/W-0h				

Table 34-76. MCAN_TXEFA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-0	EFAI	R/W	0h	Event FIFO Acknowledge Index. After the Host has read an element or a sequence of elements from the Tx Event FIFO it has to write the index of the last element read from Tx Event FIFO to EFAI. This will set the Tx Event FIFO Get Index TXEFS.EFGI to EFAI + 1 and update the Event FIFO Fill Level TXEFS.EFFL. Reset type: SYSRSn

34.7.4 MCAN_ERROR_REGS Registers

Table 34-77 lists the memory-mapped registers for the MCAN_ERROR_REGS registers. All register offset addresses not listed in Table 34-77 should be considered as reserved locations and the register contents should not be modified.

Table 34-77. MCAN_ERROR_REGS Registers

Offset	Acronym	Register Name	Protection
0h	MCANERR_REV	MCAN Error Aggregator Revision Register	
8h	MCANERR_VECTOR	MCAN ECC Vector Register	
Ch	MCANERR_STAT	MCAN Error Misc Status	
10h	MCANERR_WRAP_REV	MCAN ECC Wrapper Revision Register	
14h	MCANERR_CTRL	MCAN ECC Control	
18h	MCANERR_ERR_CTRL1	MCAN ECC Error Control 1 Register	
1Ch	MCANERR_ERR_CTRL2	MCAN ECC Error Control 2 Register	
20h	MCANERR_ERR_STAT1	MCAN ECC Error Status 1 Register	
24h	MCANERR_ERR_STAT2	MCAN ECC Error Status 2 Register	
28h	MCANERR_ERR_STAT3	MCAN ECC Error Status 3 Register	
3Ch	MCANERR_SEC_EOI	MCAN Single Error Corrected End of Interrupt Register	
40h	MCANERR_SEC_STATUS	MCAN Single Error Corrected Interrupt Status Register	
80h	MCANERR_SEC_ENABLE_SET	MCAN Single Error Corrected Interrupt Enable Set Register	
C0h	MCANERR_SEC_ENABLE_CLR	MCAN Single Error Corrected Interrupt Enable Clear Register	
13Ch	MCANERR_DED_EOI	MCAN Double Error Detected End of Interrupt Register	
140h	MCANERR_DED_STATUS	MCAN Double Error Detected Interrupt Status Register	
180h	MCANERR_DED_ENABLE_SET	MCAN Double Error Detected Interrupt Enable Set Register	
1C0h	MCANERR_DED_ENABLE_CLR	MCAN Double Error Detected Interrupt Enable Clear Register	
200h	MCANERR_AGGR_ENABLE_SET	MCAN Error Aggregator Enable Set Register	
204h	MCANERR_AGGR_ENABLE_CLR	MCAN Error Aggregator Enable Clear Register	
208h	MCANERR_AGGR_STATUS_SET	MCAN Error Aggregator Status Set Register	
20Ch	MCANERR_AGGR_STATUS_CLR	MCAN Error Aggregator Status Clear Register	

Complex bit access types are encoded to fit into small table cells. Table 34-78 shows the codes that are used for access types in this section.

Table 34-78. MCAN_ERROR_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set

Table 34-78. MCAN_ERROR_REGS Access Type Codes (continued)

Access Type	Code	Description
WD	W D	Write Decrement. Decrements the specified bit field by the amount written.
WI	W I	Write Increment. Increments the specified bit field by the amount written.
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

34.7.4.1 MCANERR_REV Register (Offset = 0h) [Reset = 66A0EA00h]

MCANERR_REV is shown in Figure 34-82 and described in Table 34-79.

Return to the Summary Table.

MCAN Error Aggregator Revision Register

Figure 34-82. MCANERR_REV Register

31	30	29	28	27	26	25	24
SCHEME		RESERVED		MODULE_ID			
R-1h		R-2h		R-6A0h			
23	22	21	20	19	18	17	16
MODULE_ID							
R-6A0h							
15	14	13	12	11	10	9	8
RESERVED					REVMAJ		
R-1Dh					R-2h		
7	6	5	4	3	2	1	0
RESERVED		REVMIN					
R-0h		R-0h					

Table 34-79. MCANERR_REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	PID Register Scheme Reset type: SYSRSn
29-28	RESERVED	R	2h	Reserved
27-16	MODULE_ID	R	6A0h	Module Identification Number Reset type: SYSRSn
15-11	RESERVED	R	1Dh	Reserved
10-8	REVMAJ	R	2h	Major Revision of the Error Aggregator Reset type: SYSRSn
7-6	RESERVED	R	0h	Reserved
5-0	REVMIN	R	0h	Minor Revision of the Error Aggregator Reset type: SYSRSn

34.7.4.2 MCANERR_VECTOR Register (Offset = 8h) [Reset = 00000000h]

MCANERR_VECTOR is shown in Figure 34-83 and described in Table 34-80.

Return to the Summary Table.

Each error detection and correction (EDC) controller has a bank of error registers (offsets 0x10 - 0x3B) associated with it. These registers are accessed via an internal serial bus (SVBUS). To access them through the ECC aggregator the controller ID desired must be written to the ECC_VECTOR field, together with the RD_SVBUS trigger and RD_SVBUS_ADDRESS bit field. This initiates the serial read which consummates by setting the RD_SVBUS_DONE bit. At this point the addressed register may be read by a normal CPU read of the appropriate offset address.

Figure 34-83. MCANERR_VECTOR Register

31	30	29	28	27	26	25	24
RESERVED							RD_SVBUS_DONE
R-0h							R-0h
23	22	21	20	19	18	17	16
RD_SVBUS_ADDRESS							
R/W-0h							
15	14	13	12	11	10	9	8
RD_SVBUS	RESERVED				ECC_VECTOR		
R-0/W1S-0h	R-0h				R/W-0h		
7	6	5	4	3	2	1	0
ECC_VECTOR							
R/W-0h							

Table 34-80. MCANERR_VECTOR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	RD_SVBUS_DONE	R	0h	Read Completion Flag Reset type: SYSRSn
23-16	RD_SVBUS_ADDRESS	R/W	0h	Read Address Offset Reset type: SYSRSn
15	RD_SVBUS	R-0/W1S	0h	Read Trigger Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10-0	ECC_VECTOR	R/W	0h	ECC RAM ID. Each error detection and correction (EDC) controller has a bank of error registers (offsets 0x10 - 0x3B) associated with it. These registers are accessed via an internal serial bus (SVBUS). To access them through the ECC aggregator the controller ID desired must be written to the ECC_VECTOR field, together with the RD_SVBUS trigger and RD_SVBUS_ADDRESS bit field. This initiates the serial read which consummates by setting the RD_SVBUS_DONE bit. At this point the addressed register may be read by a normal CPU read of the appropriate offset address. 0x000 Message RAM ECC controller is selected Others Reserved (do not use) Subsequent writes through the SVBUS (offsets 0x10 - 0x3B) have a delayed completion. To avoid conflicts, perform a read back of a register within this range after writing. Reset type: SYSRSn

34.7.4.3 MCANERR_STAT Register (Offset = Ch) [Reset = 00000002h]

MCANERR_STAT is shown in Figure 34-84 and described in Table 34-81.

Return to the Summary Table.

MCAN Error Misc Status

Figure 34-84. MCANERR_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					NUM_RAMS										
R-0h																					R-2h										

Table 34-81. MCANERR_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	Reserved
10-0	NUM_RAMS	R	2h	Number of RAMs. Number of ECC RAMs serviced by the aggregator. Reset type: SYSRSn

34.7.4.4 MCANERR_WRAP_REV Register (Offset = 10h) [Reset = 66A42A02h]

MCANERR_WRAP_REV is shown in Figure 34-85 and described in Table 34-82.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

Figure 34-85. MCANERR_WRAP_REV Register

31	30	29	28	27	26	25	24
SCHEME		RESERVED		MODULE_ID			
R-1h		R-2h		R-6A4h			
23	22	21	20	19	18	17	16
MODULE_ID							
R-6A4h							
15	14	13	12	11	10	9	8
RESERVED					REVMAJ		
R-5h					R-2h		
7	6	5	4	3	2	1	0
RESERVED		REVMIN					
R-0h		R-2h					

Table 34-82. MCANERR_WRAP_REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	PID Register Scheme Reset type: SYSRSn
29-28	RESERVED	R	2h	Reserved
27-16	MODULE_ID	R	6A4h	Module Identification Number Reset type: SYSRSn
15-11	RESERVED	R	5h	Reserved
10-8	REVMAJ	R	2h	Major Revision of the Error Aggregator Reset type: SYSRSn
7-6	RESERVED	R	0h	Reserved
5-0	REVMIN	R	2h	Minor Revision of the Error Aggregator Reset type: SYSRSn

34.7.4.5 MCANERR_CTRL Register (Offset = 14h) [Reset = 00000187h]

MCANERR_CTRL is shown in Figure 34-86 and described in Table 34-83.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

Figure 34-86. MCANERR_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							CHECK_SVBUS_TIMEOUT
R-0h							R/W-1h
7	6	5	4	3	2	1	0
RESERVED	ERROR_ONCE	FORCE_N_ROW	FORCE_DED	FORCE_SEC	ENABLE_RMW	ECC_CHECK	ECC_ENABLE
R/W-1h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-1h

Table 34-83. MCANERR_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	Reserved
8	CHECK_SVBUS_TIMEOUT	R/W	1h	Enables Serial VBUS timeout mechanism Reset type: SYSRSn
7	RESERVED	R/W	1h	Reserved
6	ERROR_ONCE	R/W	0h	If this bit is set, the FORCE_SEC/FORCE_DED will inject an error to the specified row only once. The FORCE_SEC bit will be cleared once a writeback happens. If writeback is not enabled, this error will be cleared the cycle following the read when the data is corrected. For double-bit errors, the FORCE_DED bit will be cleared the cycle following the double-bit error. Any subsequent reads will not force an error. Reset type: SYSRSn
5	FORCE_N_ROW	R/W	0h	Enable single/double-bit error on the next RAM read, regardless of the MCANERR_ERR_CTRL1.ECC_ROW setting. For write through mode, this applies to writes as well as reads. Reset type: SYSRSn
4	FORCE_DED	R/W	0h	Force double-bit error. Cleared the cycle following the error if ERROR_ONCE is asserted. For write through mode, this applies to writes as well as reads. MCANERR_ERR_CTRL1 and MCANERR_ERR_CTRL2 should be configured prior to setting this bit. Reset type: SYSRSn
3	FORCE_SEC	R/W	0h	Force single-bit error. Cleared on a writeback or the cycle following the error if ERROR_ONCE is asserted. For write through mode, this applies to writes as well as reads. MCANERR_ERR_CTRL1 and MCANERR_ERR_CTRL2 should be configured prior to setting this bit. Reset type: SYSRSn

Table 34-83. MCANERR_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	ENABLE_RMW	R/W	1h	Enable read-modify-write on partial word writes Reset type: SYSRSn
1	ECC_CHECK	R/W	1h	Enable ECC Check. ECC is completely bypassed if both ECC_ENABLE and ECC_CHECK are '0'. Reset type: SYSRSn
0	ECC_ENABLE	R/W	1h	Enable ECC Generation Reset type: SYSRSn

34.7.4.6 MCANERR_ERR_CTRL1 Register (Offset = 18h) [Reset = 00000000h]

MCANERR_ERR_CTRL1 is shown in Figure 34-87 and described in Table 34-84.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

Figure 34-87. MCANERR_ERR_CTRL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC_ROW																															
R/W-0h																															

Table 34-84. MCANERR_ERR_CTRL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECC_ROW	R/W	0h	Row address where FORCE_SEC or FORCE_DED needs to be applied. This is ignored if FORCE_N_ROW is set. Reset type: SYSRSn

34.7.4.7 MCANERR_ERR_CTRL2 Register (Offset = 1Ch) [Reset = 00000000h]

MCANERR_ERR_CTRL2 is shown in Figure 34-88 and described in Table 34-85.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

Figure 34-88. MCANERR_ERR_CTRL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC_BIT2																ECC_BIT1															
R/W-0h																R/W-0h															

Table 34-85. MCANERR_ERR_CTRL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ECC_BIT2	R/W	0h	Second column/data bit that needs to be flipped when FORCE_DED is set Reset type: SYSRSn
15-0	ECC_BIT1	R/W	0h	Column/Data bit that needs to be flipped when FORCE_SEC or FORCE_DED is set Reset type: SYSRSn

34.7.4.8 MCANERR_ERR_STAT1 Register (Offset = 20h) [Reset = 00000000h]

MCANERR_ERR_STAT1 is shown in Figure 34-89 and described in Table 34-86.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

Figure 34-89. MCANERR_ERR_STAT1 Register

31	30	29	28	27	26	25	24
ECC_BIT1							
R-0h							
23	22	21	20	19	18	17	16
ECC_BIT1							
R-0h							
15	14	13	12	11	10	9	8
CLR_CTRL_REG_ERROR	RESERVED		CLR_ECC_OTHER	CLR_ECC_DED		CLR_ECC_SEC	
R/W1S-0h	R/WD-0h		R/W1C-0h	R/WD-0h		R/WD-0h	
7	6	5	4	3	2	1	0
CTRL_REG_ERROR	RESERVED		ECC_OTHER	ECC_DED		ECC_SEC	
R/W1S-0h	R/WI-0h		R/W1S-0h	R/WI-0h		R/WI-0h	

Table 34-86. MCANERR_ERR_STAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ECC_BIT1	R	0h	ECC Error Bit Position. Indicates the bit position in the RAM data that is in error on an SEC error. Only valid on an SEC error. 0 Bit 0 is in error 1 Bit 1 is in error 2 Bit 2 is in error 3 Bit 3 is in error ... 31 Bit 31 is in error >32 Invalid Reset type: SYSRSn
15	CLR_CTRL_REG_ERROR	R/W1S	0h	Writing a '1' clears the CTRL_REG_ERROR bit Reset type: SYSRSn
14-13	RESERVED	R/WD	0h	Reserved
12	CLR_ECC_OTHER	R/W1C	0h	Writing a '1' clears the ECC_OTHER bit. Reset type: SYSRSn
11-10	CLR_ECC_DED	R/WD	0h	Clear ECC_DED. A write of a non-zero value to this bit field decrements the ECC_DED bit field by the value provided. Reset type: SYSRSn
9-8	CLR_ECC_SEC	R/WD	0h	Clear ECC_SEC. A write of a non-zero value to this bit field decrements the ECC_SEC bit field by the value provided. Reset type: SYSRSn
7	CTRL_REG_ERROR	R/W1S	0h	Control Register Error. A bit field in the control register is in an ambiguous state. This means that the redundancy registers have detected a state where not all values are the same and has defaulted to the reset state. S/W needs to re-write these registers to a known state. A write of 1 will set this interrupt flag. Reset type: SYSRSn
6-5	RESERVED	R/WI	0h	Reserved

Table 34-86. MCANERR_ERR_STAT1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ECC_OTHER	R/W1S	0h	SEC While Writeback Error Status 0 No SEC error while writeback pending 1 Indicates that successive single-bit errors have occurred while a writeback is still pending Reset type: SYSRSn
3-2	ECC_DED	R/WI	0h	Double Bit Error Detected Status. A 2-bit saturating counter of the number of DED errors that have occurred since last cleared. 0 No double-bit error detected 1 One double-bit error was detected 2 Two double-bit errors were detected 3 Three double-bit errors were detected A write of a non-zero value to this bit field increments it by the value provided. Reset type: SYSRSn
1-0	ECC_SEC	R/WI	0h	Single Bit Error Corrected Status. A 2-bit saturating counter of the number of SEC errors that have occurred since last cleared. 0 No single-bit error detected 1 One single-bit error was detected and corrected 2 Two single-bit errors were detected and corrected 3 Three single-bit errors were detected and corrected A write of a non-zero value to this bit field increments it by the value provided. Reset type: SYSRSn

34.7.4.9 MCANERR_ERR_STAT2 Register (Offset = 24h) [Reset = 00000000h]

MCANERR_ERR_STAT2 is shown in Figure 34-90 and described in Table 34-87.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

Figure 34-90. MCANERR_ERR_STAT2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC_ROW																															
R-0h																															

Table 34-87. MCANERR_ERR_STAT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECC_ROW	R	0h	Indicates the row address where the single or double-bit error occurred. This value is address offset/4. Reset type: SYSRSn

34.7.4.10 MCANERR_ERR_STAT3 Register (Offset = 28h) [Reset = 00000000h]

MCANERR_ERR_STAT3 is shown in Figure 34-91 and described in Table 34-88.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

Figure 34-91. MCANERR_ERR_STAT3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						CLR_SVBUS_TIMEOUT	RESERVED
R-0h						R-0/W1C-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED						SVBUS_TIMEOUT	WB_PEND
R-0h						R-0/W1S-0h	R-0h

Table 34-88. MCANERR_ERR_STAT3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	Reserved
9	CLR_SVBUS_TIMEOUT	R-0/W1C	0h	Write 1 to clear the Serial VBUS Timeout Flag Reset type: SYSRSn
8-2	RESERVED	R	0h	Reserved
1	SVBUS_TIMEOUT	R-0/W1S	0h	Serial VBUS Timeout Flag. Write 1 to set. Reset type: SYSRSn
0	WB_PEND	R	0h	Delayed Write Back Pending Status 0 No write back pending 1 An ECC data correction write back is pending Reset type: SYSRSn

34.7.4.11 MCANERR_SEC_EOI Register (Offset = 3Ch) [Reset = 00000000h]

MCANERR_SEC_EOI is shown in Figure 34-92 and described in Table 34-89.

Return to the Summary Table.

MCAN Single Error Corrected End of Interrupt Register

Figure 34-92. MCANERR_SEC_EOI Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							EOI_WR
R-0h							R-0/W1S-0h

Table 34-89. MCANERR_SEC_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	EOI_WR	R-0/W1S	0h	Write to this register indicates that software has acknowledged the pending interrupt and the next interrupt can be sent to the host. Note that a write to the MCANERR_ERR_STAT1.CLR_ECC_SEC goes through the SVBUS and has a delayed completion. To avoid an additional interrupt, read the MCANERR_ERR_STAT1 register back prior to writing to this bit field. Reset type: SYSRSn

34.7.4.12 MCANERR_SEC_STATUS Register (Offset = 40h) [Reset = 00000000h]

MCANERR_SEC_STATUS is shown in Figure 34-93 and described in Table 34-90.

Return to the Summary Table.

MCAN Single Error Corrected Interrupt Status Register

Figure 34-93. MCANERR_SEC_STATUS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	MSGMEM_PEND
R-0h						R-0/W1S-0h	R-0/W1S-0h

Table 34-90. MCANERR_SEC_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	RESERVED	R-0/W1S	0h	Reserved
0	MSGMEM_PEND	R-0/W1S	0h	Message RAM SEC Interrupt Pending 0 No SEC interrupt is pending 1 SEC interrupt is pending Reset type: SYSRSn

34.7.4.13 MCANERR_SEC_ENABLE_SET Register (Offset = 80h) [Reset = 00000000h]

MCANERR_SEC_ENABLE_SET is shown in Figure 34-94 and described in Table 34-91.

Return to the Summary Table.

MCAN Single Error Corrected Interrupt Enable Set Register

Figure 34-94. MCANERR_SEC_ENABLE_SET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	MSGMEM_ENABLE_SET
R-0h						R/W1S-0h	R/W1S-0h

Table 34-91. MCANERR_SEC_ENABLE_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	RESERVED	R/W1S	0h	Reserved
0	MSGMEM_ENABLE_SET	R/W1S	0h	Message RAM SEC Interrupt Pending Enable Set. Writing a 1 to this bit enables the Message RAM SEC error interrupts. Writing a 0 has no effect. Reads return the corresponding enable bit's current value. Reset type: SYSRSn

34.7.4.14 MCANERR_SEC_ENABLE_CLR Register (Offset = C0h) [Reset = 00000000h]

MCANERR_SEC_ENABLE_CLR is shown in Figure 34-95 and described in Table 34-92.

Return to the Summary Table.

MCAN Single Error Corrected Interrupt Enable Clear Register

Figure 34-95. MCANERR_SEC_ENABLE_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	MSGMEM_ENABLE_CLR
R-0h						R/W1C-0h	R/W1C-0h

Table 34-92. MCANERR_SEC_ENABLE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	RESERVED	R/W1C	0h	Reserved
0	MSGMEM_ENABLE_CLR	R/W1C	0h	Message RAM SEC Interrupt Pending Enable Clear. Writing a 1 to this bit disables the Message RAM SEC error interrupts. Writing a 0 has no effect. Reads return the corresponding enable bit's current value. Reset type: SYSRSn

34.7.4.15 MCANERR_DED_EOI Register (Offset = 13Ch) [Reset = 00000000h]

MCANERR_DED_EOI is shown in Figure 34-96 and described in Table 34-93.

Return to the Summary Table.

MCAN Double Error Detected End of Interrupt Register

Figure 34-96. MCANERR_DED_EOI Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							EOI_WR
R-0h							R-0/W1S-0h

Table 34-93. MCANERR_DED_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	EOI_WR	R-0/W1S	0h	Write to this register indicates that software has acknowledged the pending interrupt and the next interrupt can be sent to the host. Note that a write to the MCANERR_ERR_STAT1.CLR_ECC_DED goes through the SVBUS and has a delayed completion. To avoid an additional interrupt, read the MCANERR_ERR_STAT1 register back prior to writing to this bit field. Reset type: SYSRSn

34.7.4.16 MCANERR_DED_STATUS Register (Offset = 140h) [Reset = 00000000h]

MCANERR_DED_STATUS is shown in Figure 34-97 and described in Table 34-94.

Return to the Summary Table.

MCAN Double Error Detected Interrupt Status Register

Figure 34-97. MCANERR_DED_STATUS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	MSGMEM_PEND
R-0h						R-0/W1S-0h	R-0/W1S-0h

Table 34-94. MCANERR_DED_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	RESERVED	R-0/W1S	0h	Reserved
0	MSGMEM_PEND	R-0/W1S	0h	Message RAM DED Interrupt Pending 0 No DED interrupt is pending 1 DED interrupt is pending Reset type: SYSRSn

34.7.4.17 MCANERR_DED_ENABLE_SET Register (Offset = 180h) [Reset = 00000000h]

MCANERR_DED_ENABLE_SET is shown in Figure 34-98 and described in Table 34-95.

Return to the Summary Table.

MCAN Double Error Detected Interrupt Enable Set Register

Figure 34-98. MCANERR_DED_ENABLE_SET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	MSGMEM_ENABLE_SET
R-0h						R/W1S-0h	R/W1S-0h

Table 34-95. MCANERR_DED_ENABLE_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	RESERVED	R/W1S	0h	Reserved
0	MSGMEM_ENABLE_SET	R/W1S	0h	Message RAM DED Interrupt Pending Enable Set. Writing a 1 to this bit enables the Message RAM DED error interrupts. Writing a 0 has no effect. Reads return the corresponding enable bit's current value. Reset type: SYSRSn

34.7.4.18 MCANERR_DED_ENABLE_CLR Register (Offset = 1C0h) [Reset = 00000000h]

MCANERR_DED_ENABLE_CLR is shown in Figure 34-99 and described in Table 34-96.

Return to the Summary Table.

MCAN Double Error Detected Interrupt Enable Clear Register

Figure 34-99. MCANERR_DED_ENABLE_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	MSGMEM_ENABLE_CLR
R-0h						R/W1C-0h	R/W1C-0h

Table 34-96. MCANERR_DED_ENABLE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	RESERVED	R/W1C	0h	Reserved
0	MSGMEM_ENABLE_CLR	R/W1C	0h	Message RAM DED Interrupt Pending Enable Clear. Writing a 1 to this bit disables the Message RAM DED error interrupts. Writing a 0 has no effect. Reads return the corresponding enable bit's current value. Reset type: SYSRSn

34.7.4.19 MCANERR_AGGR_ENABLE_SET Register (Offset = 200h) [Reset = 00000000h]

MCANERR_AGGR_ENABLE_SET is shown in Figure 34-100 and described in Table 34-97.

Return to the Summary Table.

MCAN Error Aggregator Enable Set Register

Figure 34-100. MCANERR_AGGR_ENABLE_SET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						ENABLE_TIMEOUT_SET	ENABLE_PARITY_SET
R-0h						R/W1S-0h	R/W1S-0h

Table 34-97. MCANERR_AGGR_ENABLE_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	ENABLE_TIMEOUT_SET	R/W1S	0h	Write 1 to enable timeout errors. Reads return the corresponding enable bit's current value. Reset type: SYSRSn
0	ENABLE_PARITY_SET	R/W1S	0h	Write 1 to enable parity errors. Reads return the corresponding enable bit's current value. Reset type: SYSRSn

34.7.4.20 MCANERR_AGGR_ENABLE_CLR Register (Offset = 204h) [Reset = 00000000h]

MCANERR_AGGR_ENABLE_CLR is shown in Figure 34-101 and described in Table 34-98.

Return to the Summary Table.

MCAN Error Aggregator Enable Clear Register

Figure 34-101. MCANERR_AGGR_ENABLE_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						ENABLE_TIMEOUT_CLR	ENABLE_PARITY_CLR
R-0h						R/W1C-0h	R/W1C-0h

Table 34-98. MCANERR_AGGR_ENABLE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	ENABLE_TIMEOUT_CLR	R/W1C	0h	Write 1 to disable timeout errors. Reads return the corresponding enable bit's current value. Reset type: SYSRSn
0	ENABLE_PARITY_CLR	R/W1C	0h	Write 1 to disable parity errors. Reads return the corresponding enable bit's current value. Reset type: SYSRSn

34.7.4.21 MCANERR_AGGR_STATUS_SET Register (Offset = 208h) [Reset = 00000000h]

MCANERR_AGGR_STATUS_SET is shown in Figure 34-102 and described in Table 34-99.

Return to the Summary Table.

MCAN Error Aggregator Status Set Register

Figure 34-102. MCANERR_AGGR_STATUS_SET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				SVBUS_TIMEOUT		AGGR_PARITY_ERR	
R-0h				R/WI-0h		R/WI-0h	

Table 34-99. MCANERR_AGGR_STATUS_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-2	SVBUS_TIMEOUT	R/WI	0h	Aggregator Serial VBUS Timeout Error Status 2-bit saturating counter of the number of SVBUS timeout errors that have occurred since last cleared. 0 No timeout errors have occurred 1 One timeout error has occurred 2 Two timeout errors have occurred 3 Three timeout errors have occurred A write of a non-zero value to this bit field increments it by the value provided. Reset type: SYSRSn
1-0	AGGR_PARITY_ERR	R/WI	0h	Aggregator Parity Error Status 2-bit saturating counter of the number of parity errors that have occurred since last cleared. 0 No parity errors have occurred 1 One parity error has occurred 2 Two parity errors have occurred 3 Three parity errors have occurred A write of a non-zero value to this bit field increments it by the value provided. Reset type: SYSRSn

34.7.4.22 MCANERR_AGGR_STATUS_CLR Register (Offset = 20Ch) [Reset = 00000000h]

MCANERR_AGGR_STATUS_CLR is shown in Figure 34-103 and described in Table 34-100.

Return to the Summary Table.

MCAN Error Aggregator Status Clear Register

Figure 34-103. MCANERR_AGGR_STATUS_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				SVBUS_TIMEOUT		AGGR_PARITY_ERR	
R-0h				R/WD-0h		R/WD-0h	

Table 34-100. MCANERR_AGGR_STATUS_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-2	SVBUS_TIMEOUT	R/WD	0h	Aggregator Serial VBUS Timeout Error Status 2-bit saturating counter of the number of SVBUS timeout errors that have occurred since last cleared. 0 No timeout errors have occurred 1 One timeout error has occurred 2 Two timeout errors have occurred 3 Three timeout errors have occurred A write of a non-zero value to this bit field decrements it by the value provided. Reset type: SYSRSn
1-0	AGGR_PARITY_ERR	R/WD	0h	Aggregator Parity Error Status 2-bit saturating counter of the number of parity errors that have occurred since last cleared. 0 No parity errors have occurred 1 One parity error has occurred 2 Two parity errors have occurred 3 Three parity errors have occurred A write of a non-zero value to this bit field decrements it by the value provided. Reset type: SYSRSn

Chapter 35

EtherCAT® SubordinateDevice Controller (ESC)

This chapter describes the implementation and integration of the Ethernet for Control Automation Technology (EtherCAT®) SubordinateDevice Controller (ESC). The ESC can be mapped to any CPU subsystem (default access mapped to CPU1 subsystem), and thus any of these subsystems can run the ESC stack and application software to implement an ESC node.

On this device, the CPU1 is the main subsystem. If the intention is to use a different CPU subsystem as the owner of the ESC, then certain ESC subsystem configurations can only be done by the CPU1 during initialization before allocating the ownership of the ESC peripheral to the other CPU subsystem. These details are explained in this chapter.

Information about the EtherCAT standards and working principles can be in found these documents:

- EtherCAT ET1100 Data sheet
- EtherCAT ESC Hardware Data sheet (Section I)
- EtherCAT IP Core for Xilinx™ FPGAs

Refer to Section 35.1.11 for details on EtherCAT IP errata provided from Beckhoff ® Automation.

35.1 Introduction

Ethernet for Control Automation Technology (EtherCAT®) is an Ethernet-based fieldbus system, invented by Beckhoff ® Automation and is standardized in IEC 61158. All the SubordinateDevice (or SubDevice) nodes connected to the bus interpret, process, and modify the data addressed to them "on the fly", without having to buffer the frame inside the node. This real-time behavior, frame processing, and forwarding requirements are implemented by the EtherCAT SubDevice controller (ESC) hardware. EtherCAT does not require software interaction for data transmission inside the SubDevices. EtherCAT only defines the MAC layer while the higher layer protocols and stack are implemented in software on the microcontrollers connected to the ESC.

The EtherCAT:

- Involves MainDevice (or MDevice) and SubordinateDevice (SubDevice) setup where SubDevice nodes are physically connected
- Specializes in precise, low jitter synchronization across SubDevice nodes
- Uses IEEE 802.3 Ethernet physical layer and standard Ethernet frames

A list of relevant terms and definitions are listed in Table 35-1.

Table 35-1. Abbreviations

Name	Definition
CPU1	Main CPU1 subsystem on this MCU
CPU2	CPU2 subsystem on this MCU
CPU3	CPU3 subsystem on this MCU
DIGIO	Digital IO profile
ENI	EtherCAT Network Information
ESC	EtherCAT SubordinateDevice Controller
ESCSS or SS	Subsystem (specifically referring to EtherCAT Subsystem on this device)
ESI	EtherCAT SubordinateDevice Information
ET1100	Beckhoff EtherCAT SubordinateDevice controller
ETG	EtherCAT Technical Group
EtherCAT	Ethernet for Control Automation Technology
FMMU	Fieldbus Memory Management Units
HAL	Hardware Abstraction Layer
MII	Medium Independent Interface
NMIWDRS	NMI Watchdog Reset
PDI	Processor Data Interface
POR	Power-on-Reset
SSC	EtherCAT SubordinateDevice Stack Code
SII	SubordinateDevice Information Interface
WDRS	Watchdog Reset
XRS	External Reset

35.1.1 EtherCAT Related Collateral

Foundational Materials

- C29x Academy - Ethernet for Control Automation Technology (EtherCAT)

35.1.2 ESC Features

The ESC on this MCU provides the following functionality.

- Up to 2 MII ports to connect to EtherCAT PHYs
- Process data interface (PDI) through 16-bit asynchronous interface (ASYNC16)
- 64-bit distributed clocking.
  - Sync output signals to synchronize device events and latch input signals supporting time stamping for events.
  - Distributed clock features of SYNC0/1 (o/ps) and LATCH0/1 able to synchronize GPIOs and allow inputs from any GPIOs as well as other muxing options for internal device events.
- 8 Field bus Memory Management Units (FMMUs)
  - Supports all native types of RD/, WR/, RDWR, and built-in features of bit- and byte-addressing
- 8 Sync Managers
- I2C EEPROM interface
- Up-to 32 general-purpose inputs and 32 general-purpose outputs
- 2 SYNC and 2 LATCH signals connected to GPIO pads
- 16KB RAM with parity

35.1.3 ESC Subsystem Integrated Features

In addition to the ESC features, the following are the device-specific features provided by the integration of the ESC to the MCU:

- ESC access allocation to CPU1 subsystem, CPU2 subsystem, or CPU3 subsystem during initialization
- EtherCAT reset request from MainDevice can be routed to NMI or general interrupt controller on MCU
- RAM Parity error routed to NMI on MCU
- RTDMA access to EtherCAT RAM
- Up to 32 GPI (general-purpose inputs) and up to 32 GPO (general-purpose outputs) feature integrated in addition to 16-bit ASYNC PDI interface
- Interface to CLB
- I2C internal loopback feature for EEPROM Emulation
- Distributed clock feature of SYNC0 and SYNC1 able to synchronize PWMs, generate interrupt and RTDMA requests, trigger ECAP capture, or allow external component action through GPIO access
- EtherCAT SYNC0 and SYNC1 pulse can trigger a CLA task
- Distributed clock feature of LATCH0 and LATCH1 allowing inputs from any GPIO or PWM crossbar triggers

35.1.4 ESC versus Beckhoff ET1100

Table 35-2 details a comparison between the ESC IP Core implementation and the Beckhoff's ET1100 ESC.

Table 35-2. ESC versus Beckhoff ET1100

Feature	ESC	Beckhoff ET1100
Transmission Speed	100Mbit/s	100Mbit/s
Number of Ports	2 (MII Only)	4 (MII or EBUS)
FMMUs	8	8
SyncManagers	8	8
RAM (KByte)	16KB	8KB
Distributed Clocks	Yes, 64-bit	Yes, 64-bit
EBUS Support	No	Yes
Process Data Interfaces (PDI)	16-bit ASYNC	Digital I/O SPI 8-bit ASYNC/SYNC 16-bit ASYNC/SYNC

35.1.5 EtherCAT IP Block Diagram

Figure 35-1 shows the general functionality of etherCAT IP from Beckhoff. This chapter discusses how the etherCAT IP is integrated into the MCU.

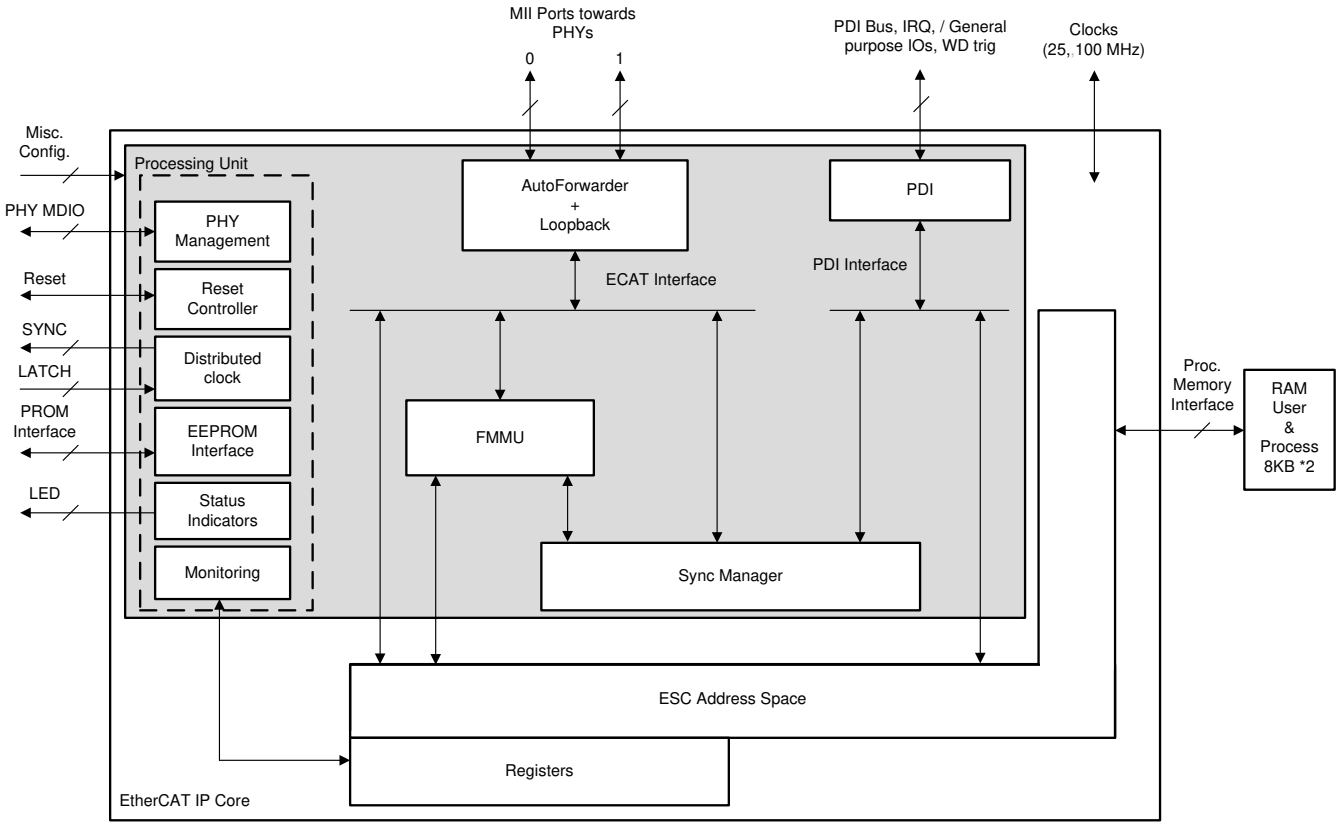


Figure 35-1. EtherCAT IP Block Diagram

35.1.6 ESC Functional Blocks

The following introduces each of the sub-blocks. For more details, refer to the Beckhoff EtherCAT documentation.

35.1.6.1 Interface to EtherCAT MainDevice

The EtherCAT MainDevice is a remote entity normally implemented through the device with ethernet switch capability. Beckhoff suggests using either a standard ethernet physical layer interface through MII or RMII protocol, or using a low-voltage differential pair signaling with EBUS protocol. Per Beckhoff specification, up to 4 PHY ports can be supported. In this MCU, implementation of ESC:

- Only MII is supported for Ethernet PHY interface
- ESC supports only 2 PHY ports

35.1.6.2 Process Data Interface

Process data interface (PDI) is a connection from the ESC IP to the microcontroller and SubordinateDevice application. The implementation of the ESC on this MCU supports a 16-bit asynchronous interface (ASYNC16) to the local host only

35.1.6.3 General-Purpose Inputs and Outputs

Since this MCU does not support digital I/O PDI interface, the ESC supports general-purpose I/O that allows 32 GPI and 32 GPO signals. These EtherCAT GPI/GPOs can be programmed to be either driven by (or drive) device I/O pins or be driven by (or drive) the same I/O pins after qualifying them using internal events. Refer to Section 35.2.9 for more details.

35.1.6.4 EtherCAT Processing Unit (EPU)

The EtherCAT Processing Unit (EPU) receives, analyzes, and processes the EtherCAT data stream. The EPU is logically located between port 0 and port 1. The main purpose of the EPU is to enable and coordinate access to the internal registers and the memory space of the ESC, which can be addressed both from the EtherCAT MainDevice and from the local application by way of the Processor Data Interface (PDI). Data exchange between the MainDevice and SubordinateDevice application is comparable to a dual-ported memory (process memory), enhanced by special functions; for example, for consistency checking (SyncManager) and data mapping (FMMU). The EPU contains the main function blocks of the EtherCAT SubordinateDevices besides Auto-Forwarding, Loop-back function, and PDI.

- **Auto-Forwarder:**
  - The Auto-forwarder function includes receiving EtherCAT frame, checksum computation and verification, timestamp capture for received frames, and forwarding to Loop-back component.
- **Loop-back Function:**
  - For the two-port implementation on this device, port 3 and port 2 are looped back by design, as shown in Figure 35-2.
  - Port 0 is a special port that is always connected to the MainDevice while designing the topology. At egress, port 0 acts as a terminal point for the network connected downstream; hence, any circulating packets get dropped at port 0 in the event of port 0 being looped-back.
  - In the event of a network or link loss downstream to the ESC port 1, the packets end in loopback.

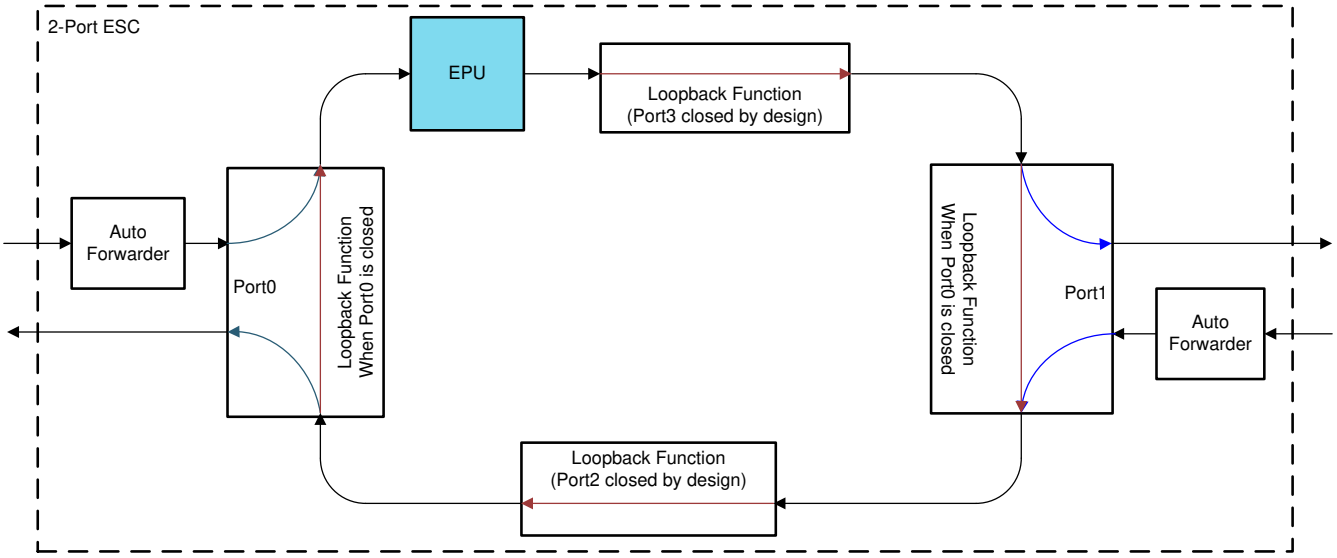


Figure 35-2. Two-port ESC Description

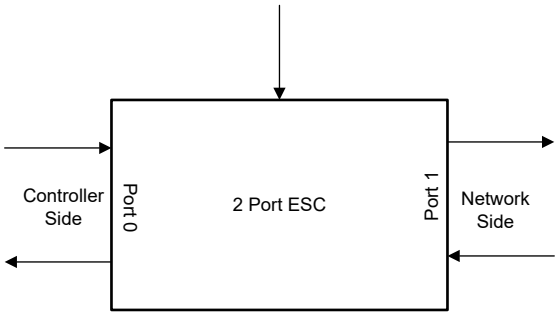


Figure 35-3. Two-port Block Diagram in EtherCAT Topology

35.1.6.5 Fieldbus Memory Management Unit (FMMU)

The fieldbus memory management unit (FMMU) maps the logical addresses of the memory elements in ESC to a physical address that allows the remote EtherCAT MainDevice to address registers and process RAM as one memory-map that can be accessed across the EtherCAT network. Mapping can be done up-to bit-wise mapping.

35.1.6.6 Sync Manager

The integrity and security of the data in dual-port memory across the MainDevice and local application is maintained by the sync manager. The sync manager allows data to be accessed as buffered or mailbox, where either there is definite read and write space or sequence defined for both of the MainDevice entities.

35.1.6.7 Monitoring

The monitoring unit contains error counters and watchdogs. The watchdogs are used for observing communication and returning to a safe state in case of an error. Error counters are used for error detection and analysis.

35.1.6.8 Reset Controller

The reset controller aggregates the internal/external resets as well as asserts the reset to external pin, which can be connected to device pin to reset companion devices on the board.

35.1.6.9 PHY Management

The PHY control is maintained through the MDIO interface which allows the configuration of the PHYs and enabling advanced features of link detection, if present, be enabled.

35.1.6.10 Distributed Clock (DC)

Distributed clocks (DC) allows the tracked time at ESC be synchronized across the EtherCAT network, which makes precisely timed events possible throughout the network. Events can be realized with triggers from ESC outputs that are programmed to create a toggle at a certain absolute time tracked by the ESC or through sampling of timestamp of certain system event inputs to ESC that can further be used for synchronizing the time.

- Refer to Section 35.2.10 for more details.

35.1.6.11 EEPROM

Every ESC requires non-volatile memory to store the configuration contents of the ESI (EtherCAT SubordinateDevice Information) that are accessed by the MainDevice normally before enabling the ESC in the network. Size of the memory is configurable with a minimum of 1K bit up to 4M bit are supported depending on the ESC. This non-volatile memory is supported in simplest form by the ESC IP core as serial access EEPROM.

- For ESI EEPROM Layout and mandatory information, refer to the ESC Hardware Data Sheet Section I - Technology (ethercat_esc_datasheet_sec1_technology) document provided by Beckhoff/ETG at www.beckhoff.com.
- EEPROM is supported using the I2C interface on this device and is further explained in Section 35.2.8.

35.1.6.12 Status / LEDs

ESC supports the status indication of application activity, link status and link errors which can be utilized for visual status indication through LEDs.

35.1.7 EtherCAT Physical Layer

ESC devices with Ethernet Physical Layer are able to support MII interfaces, RMII interfaces, and EBUS interfaces. Since RMII PHYs include TX FIFOs, the RMII PHYs increase the packet forwarding delay of an ESC device as well as the jitter. RMII as an Ethernet Physical Layer is not supported on this device due to these reasons and therefore only the MII interface is supported.

- On this device, only MII is supported (RMII and EBUS are not supported)
- Refer to the **ESC Application Note - PHY Selection Guide (an_phy_selection_guide)** provided by Beckhoff/ETG at www.beckhoff.com for additional PHY selection details.

Table 35-3 depicts the number of pins needed for MII interface for two ports.

Table 35-3. EtherCAT Physical Layer Signals

Pin	Number of Pins for 2 Ports	MII	Direction	Description
nMII_Link	2	Yes	IN	Input signal provided by the PHY, if a 100Mbps (full duplex) link is established
RX_CLK	2	Yes	IN	Receive Clock
RX_DV	2	Yes	IN	Receive Data valid
RX_DATA[1:0]	4	Yes	IN	Receive Data
RX_DATA[3:2]	4	Yes	IN	Receive Data
RX_ERR	2	Yes	IN	Receive error
TX_CLK(1)	2	Optional	IN	Transmit Clock
TX_ENA	2	Yes	OUT	Transmit Enable
TX_DATA[1:0]	4	Yes	OUT	Transmit Data
TX_DATA[3:2]	4	Yes	OUT	Transmit Data
MCLK	1	Yes	OUT	MII Interface clock
MDIO	1	Yes	BI-Directional	MII Interface Data
Total Pins:	28 (required) + 2 (optional)			

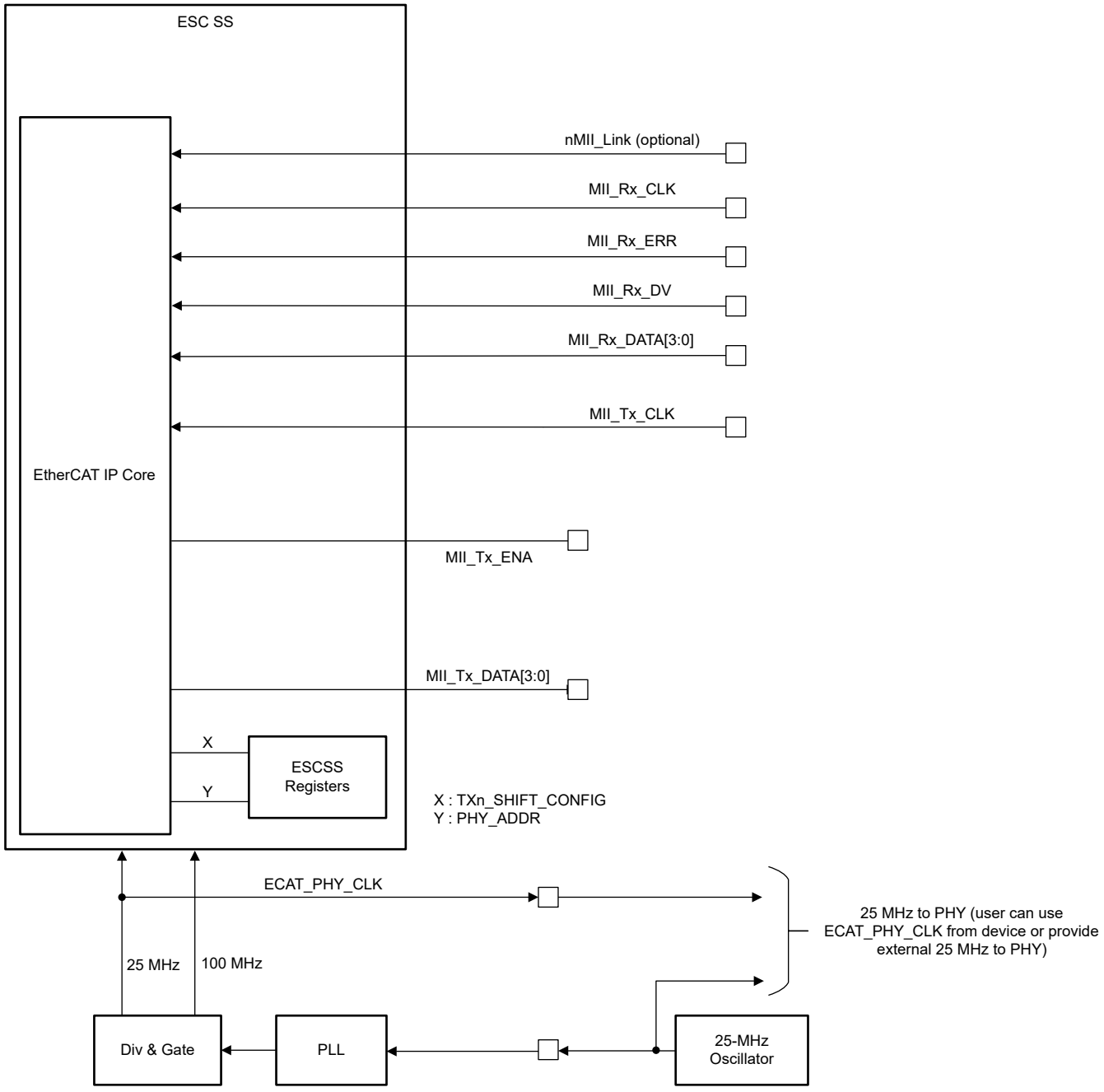
(1) The TX_CLK signal is optional when using the TX_SHIFT register for manual compensation; however, it is recommended to route the TX_CLK signals to the MII ports for improved accuracy and to avoid the need for manual compensation.

35.1.7.1 MII Interface

Ethernet IEEE802.3 specified MII interface is supported with possible minor variation in clocking to optimize the clock delay to operate the Tx FIFO. Unlike regular integration of Tx clock being sourced by the PHY, the ESC implementation allows the common source clock between PHY and ESC be used for Tx logic. The option is selected by manual Tx shift compensation which allows Tx-Data and Tx-En be compensated in steps of 10 ns to meet the timing requirements of data sampling at the PHY.

The following signals are used by the ESC to connect to an Ethernet PHY. The MDIO pins are not shown in Figure 35-4, as these pins are covered in the next section.

The MII signals TX_ERR, COL and CRS are not used by the ESC. These are not available on the MCU for EtherCAT.



Note: The PHY reset signal is also not shown in the diagram.

Figure 35-4. ESC PHY Interface Diagram

If an ESC MII interface is not used, LINK_MII has to be tied to the logic value high that indicates no link. RX_CLK, RXD, RX_ER, and especially RX_DV are connected to GND and this is taken care of by design internally when the functional IO mux for the RX pins is not configured when the IP is out of reset. The TX outputs can be left unconnected, by not configuring the Functional IO mux for respective EtherCAT functionality, unless the TX outputs are used for ESC configuration.

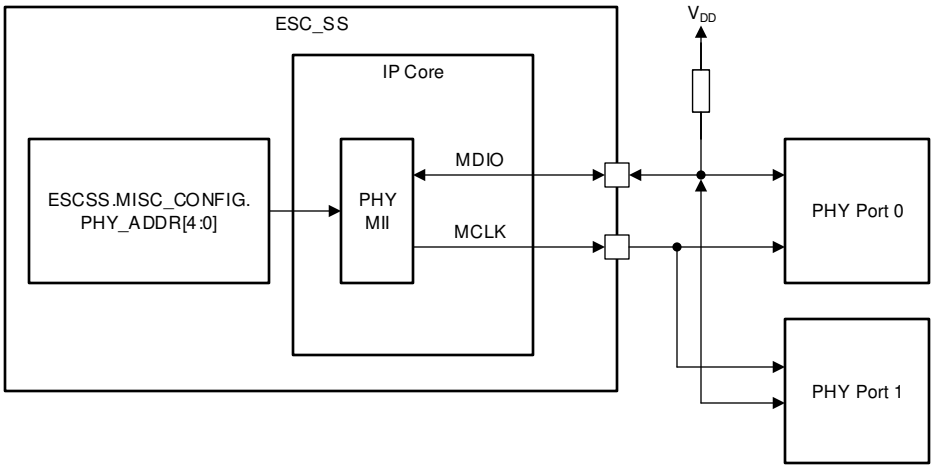
35.1.7.2 PHY Management Interface

Most ESCs with MII/RMII ports use the MII management interface for communication with the Ethernet PHYs. Most ESCs do not use the management interface for link detection or configuration of link modes. For link detection, it is recommended that the ESC use a separate signal (LINK_MII). The MII management interface can be used by the EtherCAT MainDevice – or the local MCU. Enhanced MII link detection uses the management interface for restarting auto-negotiation after communication errors occurred.

On this ESC, it is possible to make use of the MII management interface for link detection and PHY configuration.

Note

Note that the EtherCAT link status through the MII management interface option must be enabled only if a Gigabit PHY is to be supported; the EtherCAT link must not be enabled if needing to support both 10/100 and Gigabit PHY. For this reason, the link status through the MII management interface is disabled on this ESC. Instead, use the LINKACT or PHY MII_LINK signals for link status purposes.



Note: The MDIO must have a pull-up resistor (4.7 kOhm recommended) externally. MCLK is driven rail-to-rail, idle value is high.

Figure 35-5. PHY Management Interface Connectivity

35.1.7.2.1 PHY Address Configuration

The ESC typically addresses the Ethernet PHYs using the logical port number plus the PHY address offset. The Ethernet PHY addresses must correspond with the logical port number, so PHY addresses 0 and 1 are used.

A PHY address offset of 0 to 31 can be applied, which moves the PHY addresses to any consecutive address range. The ESC module expects logical port 0 to have PHY address 0 plus the PHY address offset. The PHY address offset can be selected in register ESCSS_MISC_CONFIG.PHY_ADDR[4:0].

35.1.7.2.2 PHY Reset Signal

The PHY reset signal is generated out of the ESC module. If required to release both, the PHY and ESC module synchronous out of reset, this signal can be used. Since there are no pull devices active on the MCU during and after reset, a pull down resistor must be added on this signal on the board level.

In some case, PHYs can be released from reset after releasing the ESC module. To generate a delay, the pin for nPHY_RESET can be used as an I/O and must be switched later to the alternate output function.

35.1.7.2.3 PHY Clock

The PHY Clock connectivity is shown in Figure 35-4. On this MCU, the user has an option to clock the PHY using the ESCSS_PHY_CLK signal if needed; otherwise, the user can chose to provide an external 25MHz source to the PHY and ESC (both must be clocked from the same source). For more details regarding providing PHY clock, refer to Section 35.2.6.2 and see the EtherCAT data sheet from Beckhoff.

Note

Beckhoff documents strict accuracy (ppm) requirements for the shared 25MHz ESC and PHY clock. Special care must be taken within a design to make sure the EtherCAT clocking requirements are met. Relying on external components (oscillator and clock buffer) for the 25MHz source are recommended. Validation of a design during normal operating conditions can be necessary to confirm clocking requirements are adequately met, especially if using the ESCSS_PHY_CLK signal.

35.1.8 EtherCAT Protocol

EtherCAT uses standard IEEE 802.3 Ethernet frames, thus a standard network controller can be used and no special hardware is required on the MainDevice side. EtherCAT has a reserved EtherType of 0x88A4 that distinguishes this from other Ethernet frames. Thus, EtherCAT can run in parallel to other Ethernet protocols.

- EtherCAT does not need the IP protocol; however, the EtherCAT can be encapsulated in IP/UDP.
- The ESC processes the frame in hardware; therefore, communication performance is independent from processor power.

An EtherCAT frame is subdivided into the EtherCAT frame header followed by one or more EtherCAT datagrams. At least one EtherCAT datagram has to be in the frame. Only EtherCAT frames with Type 1 in the EtherCAT Header are currently processed by the ESCs. The ESCs also support IEEE802.1Q VLAN Tags, although the VLAN Tag contents are not evaluated by the ESC.

- If the minimum Ethernet frame size requirement is not fulfilled, padding bytes are added. Otherwise, the EtherCAT frame is exactly as large as the sum of all EtherCAT datagrams plus EtherCAT frame header.
- For further reading, including details on network time or network propagation delay, refer to the ESC Hardware Data Sheet Section I - Technology (ethercat_esc_datasheet_sec1_technology) document provided by Beckhoff/ETG at www.beckhoff.com.

35.1.9 EtherCAT State Machine (ESM)

The EtherCAT State Machine (ESM) is responsible for the coordination of the MainDevice and SubordinateDevice applications at start up and during operation. State changes are typically initiated by requests from the MainDevice. The state changes are acknowledged by the local application after the associated operations have been executed. Unsolicited state changes of the local application are also possible.

There are four states (see Figure 35-6) an ESC can support, plus one optional state:

- Initialize (Init)
- Pre-Operational
- Safe-Operational
- Operational
- Bootstrap (Optional)

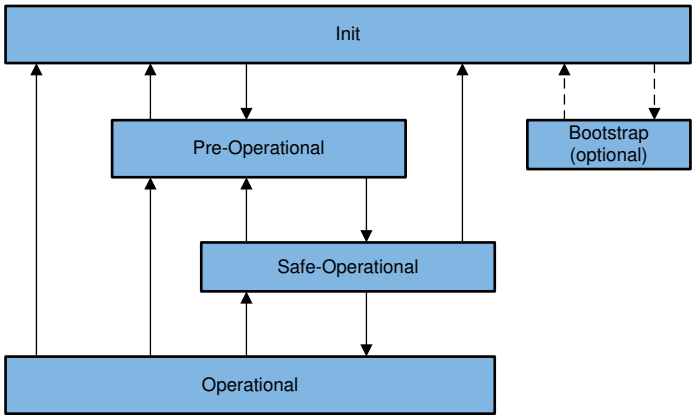


Figure 35-6. EtherCAT State Machine

Note

Not all state changes are possible, for example, the transition from Init to Operational requires the following sequence: Init → Pre-Operational → Safe-Operational → Operational.

35.1.10 More Information on EtherCAT

- For further information on EtherCAT, refer to the EtherCAT specifications, available from the EtherCAT Technology Group (ETG, www.ethercat.org).
- Refer to the IEC standard “Digital data communications for measurement and control – Fieldbus for use in industrial control systems”, IEC 61158 Type 12: EtherCAT, available from the IEC (www.iec.ch).
- Documentation on Beckhoff Automation ESC are available at the Beckhoff website (www.beckhoff.com), for example, data sheets, application notes, and ASIC pinout configuration tools.

35.1.11 Beckhoff® Automation EtherCAT IP Errata

Table 35-4 details the Beckhoff ®Automation errata of the EtherCAT IP integrated onto this device.

Table 35-4. EtherCAT IP Errata

Errata Number	Description
ER#156	WKC increment for reading reserved FMMU registers (+0xD-+0xF)
ER#158	WKC increment for writing 0x0914-0x0917
ER#162	ESC DL Control (0x0101) loop setting Auto-Close (01): if port waits for write access to be opened, the temporary loop bit (0x0100[1]) is not taken into account when a write command to ESC DL Control occurs.
ER#164	EEPROM FSM cannot accept another command within 1680 ns after finishing a previous command
ER#165	EtherCAT reset register 0x0040 cannot be written using VLAN tagged frames, because state is reset due to double ECAT_CLEAR. VLAN tagged frames are rarely used for EtherCAT.
ER#168	Changing SyncSignal cycle times during activation can lead to extremely long cycle times, resulting from (intermediate) violation of the minimum delay between two consecutive pulses. For example, changing Sync1 from a few ns before Sync0 to a few ns after Sync0 results in a missed Sync1 time, which can take up to several 32/64-bit turn-arounds until the next pulse occurs. Changing cycle time from PDI can result in usage of intermediate, inconsistent values, which can result in unwanted cycle times or extremely long delays as above.

35.2 ESC and ESCSS Description

This section details the aspects of the ESC integration in the MCU. On this MCU, the ESC is integrated such that the ESC is accessed by either the CPU1 (main) subsystem or the CPU2 or the CPU3 subsystem. The MCU with the ESC functions as an EtherCAT SubordinateDevice device.

Figure 35-7 shows the ESC on this MCU.

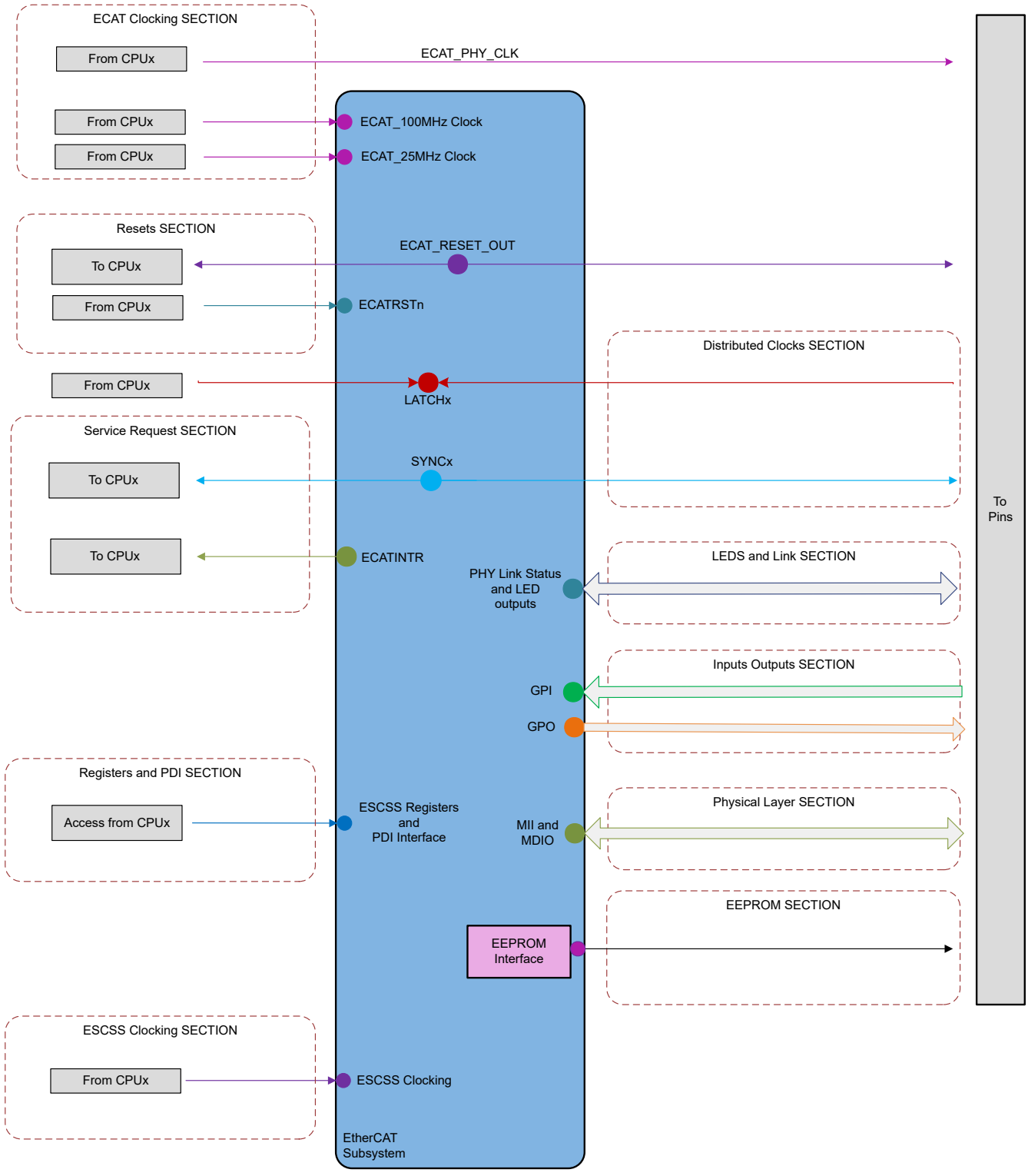


Figure 35-7. ESC Integration on MCU

Table 35-5. ESC Integration Figure Sections

ESC Figure Sections	More Information Links
ECAT/ESCSS Clocking	ESC Clocking
Resets	ESCSS Resets
Service Request	SYNC Signals Interrupts and Interrupt Mapping
Registers and PDI	ESC SubSystem
Distributed Clocks	Distributed Clocks - Sync and Latch
LEDS and Link	LED Controls
Inputs Outputs	General Purpose Inputs and Outputs
Physical Layer	EtherCAT Physical Layer
EEPROM	SubordinateDevice Node Configuration and EEPROM

There are various actions that the controlling processor (CPU1/CPU2/CPU3) needs to perform in response to actions initiated by the EtherCAT MainDevice meant for this particular SubordinateDevice . All these interactions take place through one of the following:

- The 16-bit asynchronous interface through which registers and RAM of the EtherCAT IP are read/written.
- An interrupt request going from the ESC to the Local Host (CPU1 or CPU2 or CPU3).
- SYNC0 and SYNC1 pulses generated by the ESC that are used as triggers for initiating further action.
- LATCH0 and LATCH1 that are sampled by the ESC.
- EtherCAT general-purpose inputs and outputs.

35.2.1 ESC RAM Parity and Memory Address Maps

This section details the ESC RAM parity logic, CPU1 ESC memory address map, and CPU2 ESC memory address map.

35.2.1.1 ESC RAM Parity Logic

The ESC RAM has parity logic to make sure the data integrity of the contents. The byte-wide parity is implemented, as PDI accesses to RAM can be done byte wide. To make sure that errors in parity logic itself do not mask the memory error, redundant parity generation logic is used. The parity generated by both logics is compared to make sure safe operation of the memory accesses. Use the INITIATE_MEM_INIT bit from the ESCSS register to trigger the memory initialization.

35.2.1.2 CPU1 ESC Memory Address Map

When the ESC is allocated to CPU1, Table 35-6 is the memory map and associated details.

Table 35-6. ESC Address Map on CPU1

Memory Map Region	Memory Region	Accessed Through	Parity Scheme	RTDMA Access	CLA Access	CPU2 Access	Security	Access/ Bus
DPRAM memory map (16KB)	0x0005 0800 - 0x0005 27FF	PDI interface of ESC	1 parity bit for 8 bits of data	Yes	No	No	No	Async
ESC Registers	0x0005 0000 - 0x0005 07FF	PDI Interface of ESC	No	No	No	No	No	Async
ESCSS Registers(1)	0x0005 7E00 - 0x0005 7FFF	Direct Access	No	No	No	No	No	VBUS32

(1) When register access protection is enabled, all write accesses are blocked; however, a violation interrupt is not generated in the event of an attempted write access.

35.2.1.3 CPU2 ESC Memory Address Map

When ESC is allocated to CPU2, Table 35-7 is the memory map and features.

Table 35-7. ESC Address Map on CPU2

Memory Map Region	Memory Region	Accessed Through	Parity Scheme	RTDMA Access	Bus/Protocol	Security
ESC RAM memory map (16KB)	0x400A 1000 - 0x400A 4FFF	PDI interface of ESC	1 parity bit for 8 bits of data	Yes	System/PDI	No
ESC Registers	0x400A 0000 – 0x400A 0FFF	PDI	No	No	System/PDI	No
ESC_SS Registers	0x400A FC00 – 0x400A FFFF	Direct Access	No	No	System/VBUSP	No

35.2.2 Local Host Communication

Figure 35-8 shows how the ESC and ESC subsystem connects to the local host subsystem. The local host subsystem can include the CPU (CPU1 or) of the device and the RTDMA engine as the bus controller accessing the ESC. This local host subsystem is acting upon the commands that are sent by the EtherCAT MainDevice in the form of EtherCAT datagrams.

The 16-bit asynchronous interface and the interrupt request line are the main channels through which the local host subsystem interacts with the ESC. The interrupt request line can be used to trigger actions based on ESC internal events, exception conditions or time synchronized events. The RTDMA engine is used to transfer the contents from the process data memory to system RAM on any of these events. The user application needs to select the events that act as interrupts or RTDMA requests. The mask and clear events for the interrupt causes are configured to be accessed by the CPU/Co-processor that is controlling the ESC.

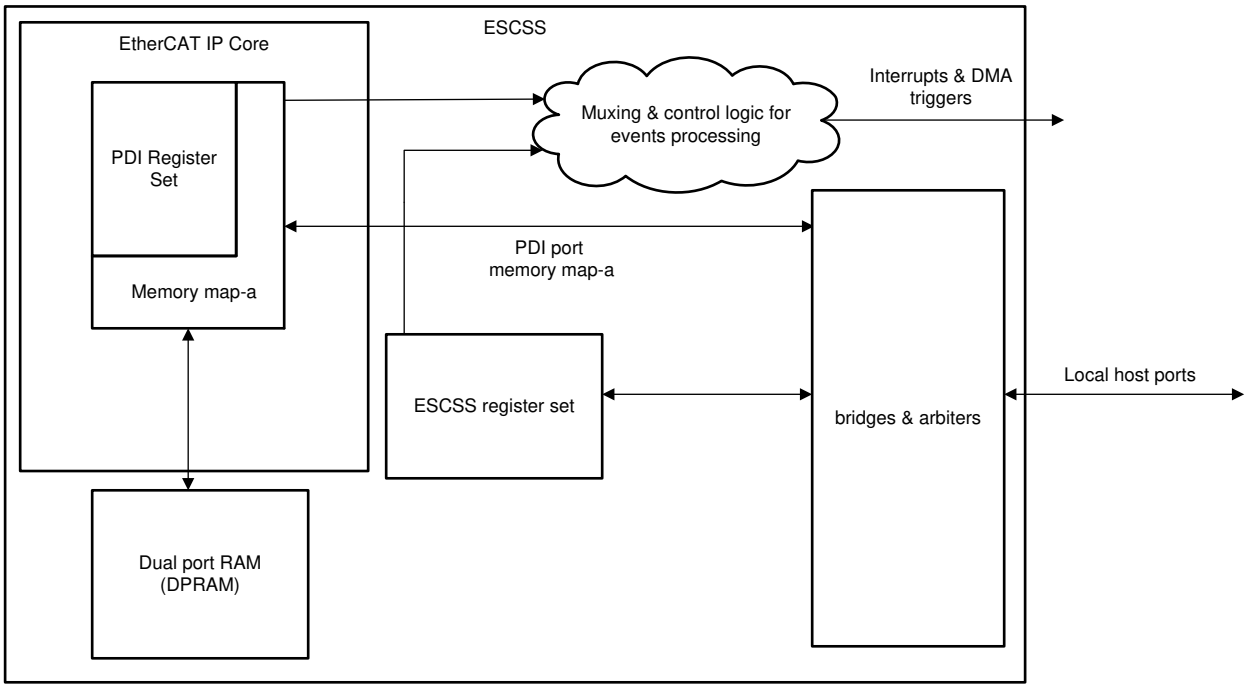


Figure 35-8. Interaction of ESCSS with the CPU Subsystem

35.2.2.1 Byte Accessibility Through PDI

The ESC has a few registers which need byte wide access to affect the functionality. The EtherCAT implementation on this MCU is as follows:

- CPUs support byte accesses

35.2.2.2 Software Details for Operation Across Clock Domains

The registers accessible from the CPU are in the system clock domain and are synchronized to the PDI clock before being used within the EtherCAT IP. Given the frequency ratios and different styles of synchronization schemes, the application needs to assume there is a delay in getting the values transferred from one domain to other. Such a delay can vary based on the frequency of system clock and type of synchronizer used in the path.

For example, a system clock of 200MHz and ESCSS running at 100MHz requires at least 10 clock-system clock cycles delay before values are affected on the other side. If a write occurs to the PDI and some other action is performed elsewhere (not involving the PDI), then the software must perform a simple read to make sure that the write is complete.

35.2.3 Debug Emulation Mode Operation

There are two aspects of the EtherCAT IP operation that need to be considered from a debug emulation stand point.

- **Debugger Writes/Reads of EtherCAT IP registers/memories Condition:** The EtherCAT IP does not have any mechanism to identify a debug initiated read/write. Debug accesses to the registers or the ESC RAM can affect the state of the EtherCAT IP. This is addressed by the following:
  - **CPU1/CPU2/CPU3:** The ENABLE_DEBUG_ACCESS bit must be set in the ESCSS Access Control Register to enable user access to the ESC RAM and EtherCAT registers for the purpose of debug. By default, this is disabled.

35.2.4 ESC SubSystem

The EtherCAT SubordinateDevice Controller SubSystem (ESCSS) wraps the ESC core with required register configurations and required logic for different functions of the EtherCAT and RAM (ESC RAM). This section covers the ESCSS integration aspects. Figure 35-9 shows the EtherCAT subsystem Integration.

- **ESCSS configuration registers interface** is the port for accessing the critical device level configurations which are a must have for the ESCSS to function.
- **ESCSS register interface** has control and status registers including SYNC, LATCH configurations, interrupt related controls, and GPIO related controls.
- **PDI Async interface** is a 16-bit wide data interface that allows the local application to access the registers internal to the EtherCAT IP Core as well as the dual-port memory (ESC RAM) through the SyncManager and the FMMUs.

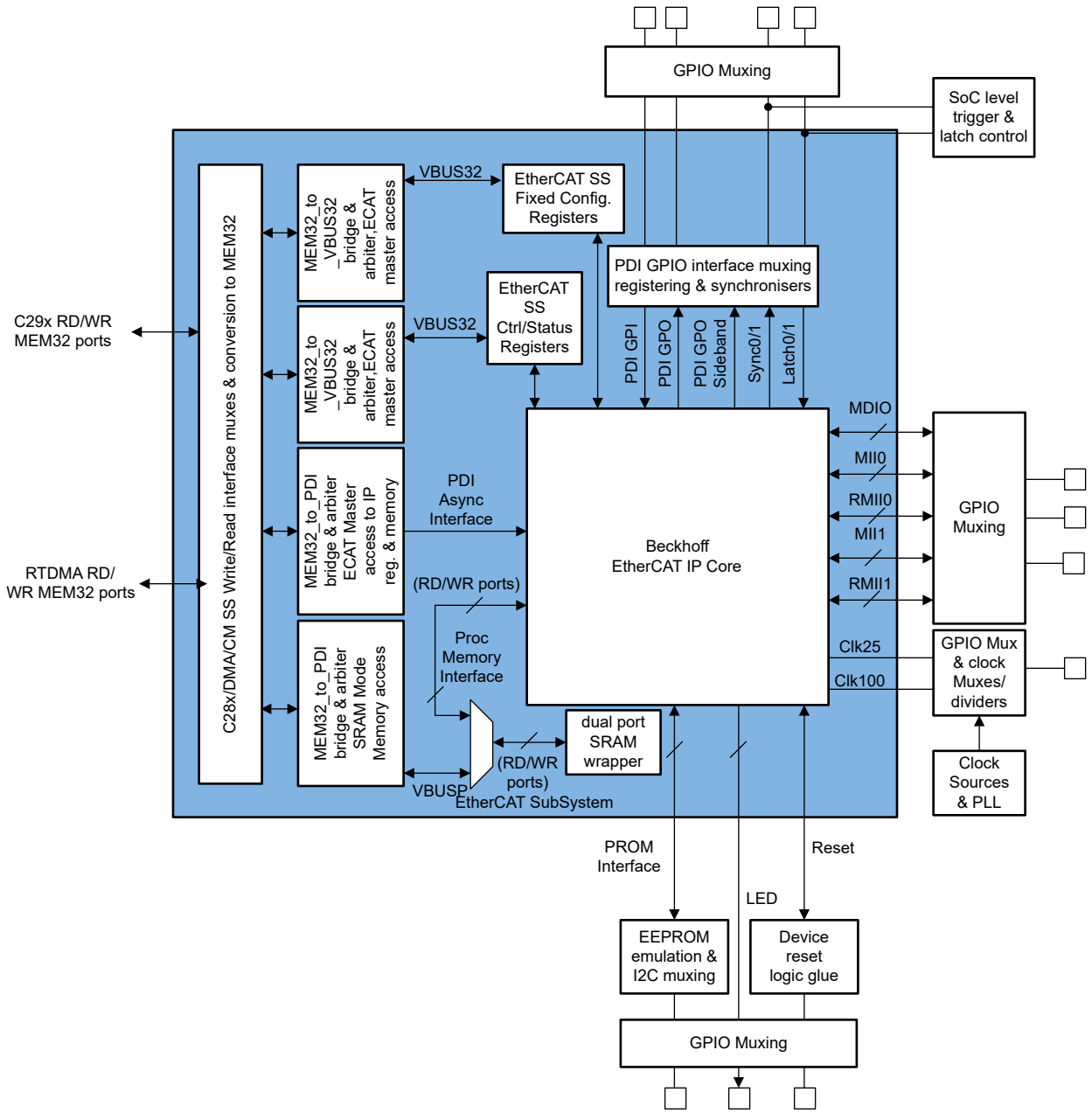


Figure 35-9. ESCSS Wrapper

35.2.4.1 CPU1 Bus Interface

Interface to the CPU1 core is retained at the ESCSS boundary as a native mem32 interface that connects the RD and WR ports independently to all of the SubordinateDevice entities within the ESCSS.

- CPU1 MEM32 RD and WR ports: Configuration Registers, Control/Status Registers, PDI
- CPU1 RTDMA MEM32 RD and WR ports: PDI (access to ESC RAM only)

35.2.4.2 CPU2/CPU3 Bus Interface

The CPU2/CPU3 interfaces involve a bus that is arbitrated between the CPU2/CPU3 and RTDMA accesses to the PDI .

- CPU2/CPU3 bus ports: Configuration Registers, Control/Status Registers, PDI

- RTDMA ports: PDI (access to ESC RAM only)

35.2.5 Interrupts and Interrupt Mapping

The ESC has one interrupt line that can be connected to the local host (CPU1 or CPU2 or CPU3) that is called PDI IRQ from the ESC. Besides this, there are other interrupt causes generated within the ESCSS. These are aggregated to a total of 4 interrupt lines that are connected to the local host core.

Table 35-8 summarizes these exceptions and the mapping. There are 4 interrupt lines provided for ESCSS which are: ECATSS_Intr, RESET_OUT_Intr, SYNC0_Intr and SYNC1_Intr. Each of these causes have an independent set of mask/clear/set controls and raw/masked interrupt status flags. This allows independent cause servicing on the exception event with flexibility to mask or service the desired set of causes. All these interrupts are mapped onto PIPE on CPU1, CPU2, and CPU3. Refer to the *System Control and Interrupts* chapter for details on the interrupt numbers.

In the event that the same exception is available to the multiple bus MainDevice cores of the local host CPU, the application software is expected to make sure that clearing of the RIS is a software synchronized event between those MainDevices and there is no stale exceptions pending for one controller while the other clears the RIS. In other words, there is no separate exception cause (RIS/MIS) copy per MainDevice but are common interrupt registers for the local host across MainDevices.

Table 35-8. Service Request Generation Map

Source	Description	MainDevice				
		CPU1	CLA	CPU1 RTDMA	CPU2/CPU3	RTDMA
EtherCAT IRQ	AL Event Request of the ESC	ECATSS_ intr	ECATSS_ intr	Not Available	ECATSS_ intr	Not Available
PDI Interface Timeout Error	PDI Interface WatchDog timeout error	ECATSS_ intr	ECATSS_ intr	Not Available	ECATSS_ intr	Not Available
EtherCAT RESET_ OUT Event	RESET_OUT can be programmed as interrupt to the CPU to either complete the reset sequence with required pre-steps if an or acknowledge the reset request in some other way. Given the high priority nature of the signal independent interrupt line is allocated	RESET_ OUT_Intr	RESET_ OUT_Intr	Not Available	RESET_ OUT_Intr	Not Available
SYNC0 Event	Precise time event 2, can be used to start routine SYNC1_Intron cores or data transfer using RTDMA. Given the precise time and priority of these interrupts a separate interrupt line is dedicated.	SYNC0_ Intr	SYNC0_ Intr	SYNC_ DMAReq	SYNC0_ Intr	SYNC_ DMAReq
SYNC1 Event	Precise time event 1, can be used to start routine on cores or data transfer using RTDMA. Given the precise time and priority of these interrupts a separate interrupt line is dedicated.	SYNC1_ Intr	SYNC1_ Intr	SYNC_ DMAReq	SYNC1_ Intr	SYNC_ DMAReq

35.2.6 Power, Clocks, and Resets

This section details the ESCSS power requirements, clocking, and resets. The EtherCAT IP has 2 clock ports and one reset output. Since the PLLs, dividers, and gating is implemented as part of system control, this setup is always handled by CPU1 during the initialization sequence.

35.2.6.1 Power

The ESC module is inside the main power domain like other peripherals and there are no special considerations about the power-up or power-down sequences that need to be taken. Refer to the *System Control and Interrupts* chapter for different power modes.

35.2.6.2 Clocking

Two functional clock inputs are defined for the ESC: CLK25 (25MHz) and CLK100 (100MHz).

EtherCAT functional clock inputs CLK25 and CLK100 are sourced by the MCU clocking module either using SYSPLL or AUXPLL. At the SoC level, you have multiple options to choose from regarding what inputs are used for the SYSPLL or AUXPLL.

- Due to the 25ppm requirement for EtherCAT, an external oscillator of 25MHz is suggested to be used as the main clock source.
- A less accurate clock than 25ppm limits the ability of the ESC to act as the network reference clock. For practical reasons, clock accuracy must be the same or better than the Ethernet clock source that is 50ppm.

Note

- CLK25 and CLK100 are used for the ESC core operation. These two clocks and the clocks to the PHY **must have a common source**, which establishes a deterministic phase relationship between PHY clocks and ESC clocks.
- In MII mode, the 25MHz RX clock is sourced by the PHY while the TX clock from the PHY is optional (which effectively saves a pin). The phase differential between the PHY TX clock and CLK25 can be compensated to the TX data and TX EN using the manual compensation mode in increments of 10ns. Connecting the TX clock avoids the need to perform the manual compensation.

Figure 35-10 shows the clocks connectivity. Refer to the *System Control and Interrupts* chapter for additional details of the PLL source selections and clock dividers.

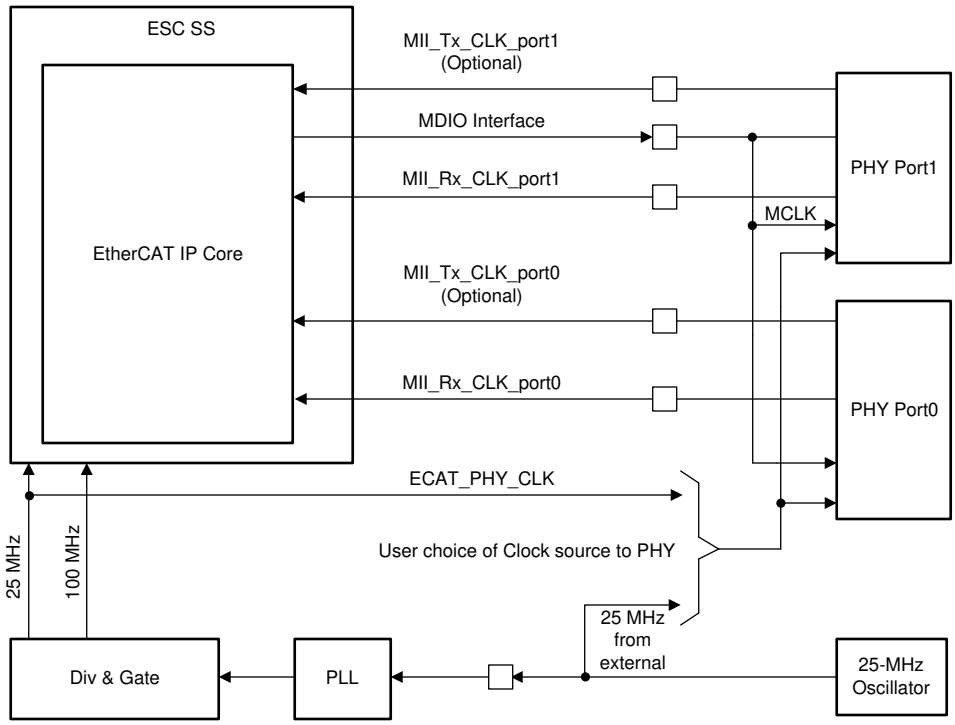


Figure 35-10. Clocking of ESC

In the case of a low-accuracy clock (up to 100ppm) being used, a clock accuracy of 25ppm cannot be feasible, then the following restrictions apply:

- RX FIFO size cannot be reduced lower than 7 (default for 100ppm), which in turn affects the latency of the transfer as the transfer starts after half of the FIFO full threshold, thus resulting in a RX Delay.
- This ECS cannot be used as the first SubordinateDevice from the MainDevice in the network, as typically the first SubordinateDevice from the MainDevice is treated as the reference clock for the distributed clocks functionality.
- The number of iterations required for synchronizing clocks (specially the drift computation) increases.

35.2.6.3 Resets

This section describes the ESCSS sources of reset. There are other conditional reset sources that can impact ESCSS and those include all the device level reset sources including: device pin reset “XRSN” or software initiated device level reset “SYSRSn”.

Note

Whenever the device comes out of reset, software must execute the following sequence to make sure that PHY can see the full stretching of the reset:

1. Configure the EtherCAT GPIOs
2. Put the ESCSS into soft reset
3. Bring the ESCSS out of soft reset

Refer to Section 35.3 more information on proper software procedure.

35.2.6.3.1 Chip-Level Reset

Chip-level resets (such as POR, XRS, WDRS, or NMIWDRS) are a MainDevice reset that resets the entire device including the ESC.

35.2.6.3.2 EtherCAT Soft Resets

ESCSS can be reset by three soft reset mechanisms, including:

- System control level software programmable (SOFTPRES23) that is reset only by chip-level and EtherCAT resets. Default keeps the ESC in reset that is released by the local application after setting the ESC ready for operation.
- Reset by a particular command sequence from the remote EtherCAT MainDevice.
- Reset by a particular command sequence from the local host application. (core only)

35.2.6.3.3 Reset Out (RESET_OUT)

Multiple reset sources of ESCSS are combined to make the reset vector that drives the EtherCAT and companion devices (PHYs) on this MCU. The RESET_OUT is asserted only when configured to be a reset. If configured as an interrupt or NMI, the local host completes the reset.

By default, RESET_OUT performs no action and must be setup to cause the reset to the EtherCAT core and PHY. The input events to RESET_OUT include the remote EtherCAT MainDevice, local host application, and EtherCAT soft reset.

When the intent is to perform a reset to the ESCSS and the device, the following options are available to reset the device after software has given a reset to the ESCSS:

- Use the system control SIMRESET register to generate a device reset.
- Use the watchdog or NMIWD to generate a device reset.

35.2.7 LED Controls

There are four LED outputs from the ESC that can be routed out of GPIO to indicate different status of the ESC. Table 35-9 is the recommendation for applications on the usage, especially for the pin sensitive systems that can support this and still meet the intent. All the LED functions are routed to the peripheral pin mux and the user must configure the LEDs as per the available pins.

Table 35-9. Status LED Options and Priority

LED	Priority Order	Function	Recommendation	Usage Priority
RUN	5	Shows the status of ESC state machine with different blinking patterns.	In the case where color LEDs are not supported, then this needs to be supported.	Must
ERR(1)	4	Indicates the Error in ESC operation.	In the case where color LEDs are not supported, then at least the Error status needs to be reported.	Must
STATE	2	Combination of the RUN and ERR LED.	Not supported on this device.	-
LINKACT0 (ESC_LED_LINK0_ACTIVE)	3	Link Active Status for link towards the MainDevice.	Link Status on MainDevice side (Port0) is important to know, if the ESC and network downstream is part of the network or not. The prior ESC loopbacks, if this node fails. The status on this LED eases the debug.	Desired
LINKACT1 (ESC_LED_LINK1_ACTIVE)	1	Link Active status for link towards Network side.	The Link status on the network side (Port 1) is critical from continuity through the ESC point; hence, this is kept highest priority status reporting.	Must

(1) Driven by software and not the ESC on this device.

The PHY MII_LINK (ESC_PHYx_LINKSTATUS) indication is an indication from PHY and does not indicate if the established link meets the characteristics including auto-negotiation as required by EtherCAT. Hence, the indication may not be a true status indication; however, it is critical to shorten reaction time in the event of link loss that is crucial for redundancy operation. Using LINKACT LEDs (LED_LINKx_ACTIVE) for the status output and MII_LINK (ESC_PHYx_LINKSTATUS) for the status input is the best usage; however, using the LEDs are not practical to sacrifice 4 pins. The tradeoff of whether MII_LINK is used as a link status LED or LINKACT is used depends on link loss reaction time requirements and available number of GPIOs for the system.

The PHY MII_LINK (ESC_PHYx_LINKSTATUS) signal is an active-low signal input to the C2000 ESC. It is possible to use the GPxINV register to invert the signal's polarity before the signal enters the ESC, if the system hardware cannot provide the desired polarity for this signal.

The LINKACT LEDs (LED_LINKx_ACTIVE) is an output signal of the ESC intended to control an LED for link indication.

Table 35-10. LINKACT and PHY MII_LINK States

Link State	LINKACTx (LED_LINKx_ACTIVE) State	PHY MII_LINK (ESC_PHYx_LINKSTATUS) State
No Link	0 (LOW)	1 (HIGH)
Link WITHOUT Activity	1 (HIGH)	0 (LOW)
Link WITH Activity	Toggling / Blinking	0 (LOW)

Note

The EtherCAT link status through the MII management interface option must be enabled if only a Gigabit PHY is to be supported and must not be enabled if needing to support both 10/100 and Gigabit PHY. For this reason, the link status through MII management interface is disabled on this ESC. Instead, use the LINKACT or PHY MII_LINK signals for link status purposes.

35.2.8 SubordinateDevice Node Configuration and EEPROM

The ESC requires non-volatile memory (for example, an EEPROM with I2C interface), which is referred to as the SubordinateDevice Information Interface (SII) memory, to store the EtherCAT SubordinateDevice Information (ESI) data.

The ESI data contains information about the SubordinateDevice device identification, supported features, and other network accessible properties.

Note

The ESI XML file is generated from Beckhoff's SubordinateDevice Stack Code (SSC) tool.

- **EtherCAT SubordinateDevice**
  - Uses the SII upon power-on or reset to load configuration data into the EtherCAT SubordinateDevice configuration registers.
- **EtherCAT MainDevice**
  - During development, the MainDevice is used to program the device SII memory with the ESI data.
  - When the ESI file isn't provided to the MainDevice, the MainDevice reads the device SII memory upon boot-up to determine the SubordinateDevice device properties such as the process data and their mapping options, the supported mailbox protocols including optional features, as well as the supported modes of synchronization.

35.2.9 General-Purpose Inputs and Outputs

The ESC subsystem implementation on this MCU supports 32 general-purpose inputs (GPI) and 32 general-purpose outputs (GPO) in addition to the PDI.

These GPI/GPOs provide an advantage for embedded applications over discrete components because the GPI/GPOs can selectively drive or sense GPIOs in a time controlled manner. For example, it can be a sensor/actuator control, status read, LED status update, and so on. The following section describes the integration and usage of this feature.

35.2.9.1 General-Purpose Inputs

General-purpose inputs are connected to the device GPIO pin mux through an input buffer with an option to directly connect the asynchronous input flowing directly into the ECS or through a ESCSS register pipeline, which can hold the value stable while the value is captured within the ESC. The state of the GPI can be captured based on the following events:

- **Start-of-Frame (SOF):** Contents can be ready for capture in the frame if requested by the MainDevice
- **SYNC0/1:** Contents captured on a precise time tick
- **LATCH0/1:** Contents synchronized with LATCHIN read (allows simultaneous input and timestamp capture)

The ESCSS_GPIN_DAT register, containing the contents of the GPIN data, is available for reading from host CPU to allow debug access. Additionally, ESCSS_GPIN_DAT allows CPU writes for being used GPIN override purposes such as in place of using GPIOs.

GPIs are divided into 4 sets: GPI0:7, GPI8:15, GPI16:23, and GPI24:31, for clocking the capture trigger, which means the same capture trigger has to be used for the IOs within a set. Thus, either a bus can be formed out of these or individual IOs which need to be aggregated under common trigger can be combined in one set. This allows limited freedom of trigger selection for inputs. Selection of which Inputs (ESCSS_GPIN_SEL) and outputs (ESCSS_GPOUT_SEL) can be connected to GPIO is possible at each single IO level.

Figure 35-11 shows the integration of the GPI feature.

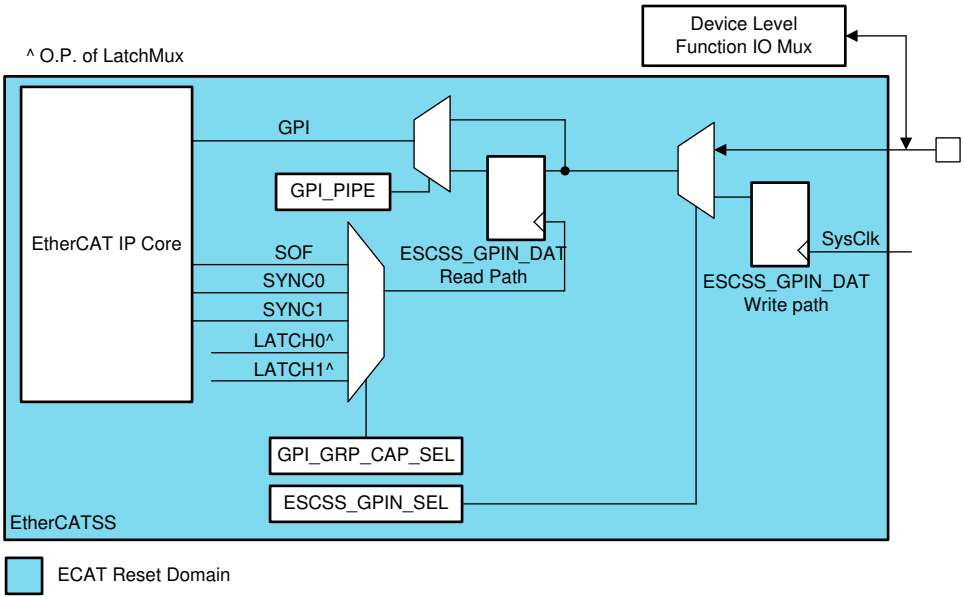


Figure 35-11. ESCSS General-Purpose Inputs Integration

Note

For software: The write data gets reflected on the ESCSS_GPIN_DAT read path only when one of the configured events, like SOF, SYNC0/1, and so on, is seen as configured after a few cycles of that event getting generated. Additionally, the copy of ESCSS_GPIN_DAT register readable by the CPU is provided without synchronization; therefore, multiple reads must be performed to confirm a stable value before using the value.

35.2.9.2 General-Purpose Output

General-purpose outputs are connected to the device GPIO pin mux through an output buffer with an option to directly connect from the ESC IP core registers to the external world or through a ESCSS register pipeline, which can hold the output value stable. The state of the GPO can be set based on the following events:

- **End-of-Frame:**Contents can be updated by the MainDevice from the last received frame
- **SYNC0/1:** Contents can be updated based on a precise time tick
- **Watchdog Trigger:** Contents can be updated based on the last successful access to process data memory

The ESCSS_GPOUT_DAT register, containing the contents of the GPO data, is available for reading from host CPU to allow debug access or to read in place of GPIOs.

GPOs are divided into 4 sets: GPI0:7, GPI8:15, GPI16:23, and GPI24:31, for clocking the output trigger, which means the same output trigger has to be used for the IOs within a set. Thus, either a bus can be formed out of these or individual IOs which need to be aggregated under common trigger can be combined in one set. This allows limited freedom of trigger selection for inputs. Selection of which inputs (ESCSS_GPIN_SEL) and outputs (ESCSS_GPOUT_SEL) can be connected to GPIO is possible at each single IO level.

Figure 35-12 shows the integration of the GPO feature.

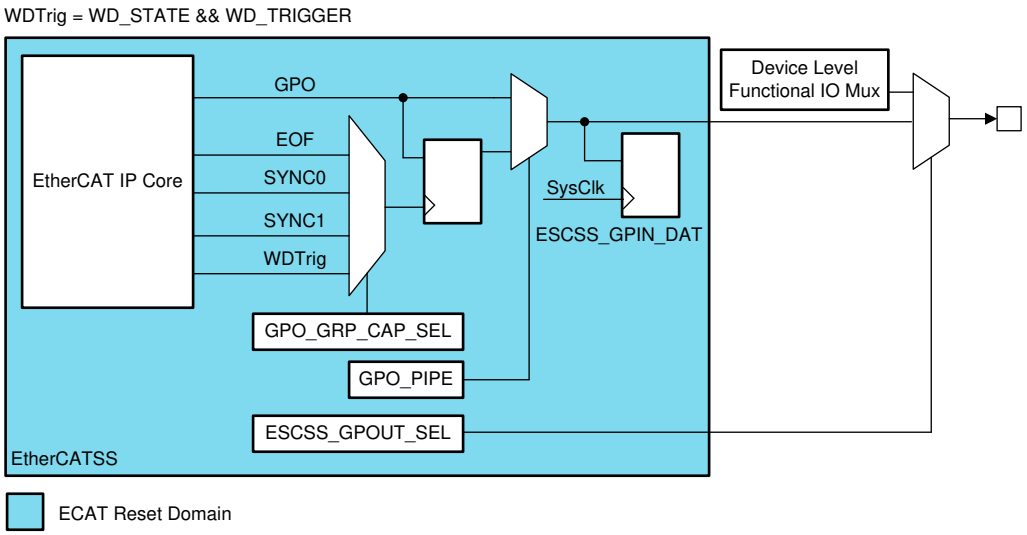


Figure 35-12. ESCSS General-Purpose Output Integration

Note

In Figure 35-12, the synchronization between CLK100 or CLK25 with respect to SysClk is not shown and is implicit by design. The local host can adjust for the synchronization delays as required while processing the data in either direction.

35.2.10 Distributed Clocks – Sync and Latch

Distributed clocks (DC) is a differentiating feature of the EtherCAT network. Distributed clocks allows all the SubordinateDevice nodes in the EtherCAT network to be bound in a tight margin of time to synchronize the events and the EtherCAT MainDevice command.

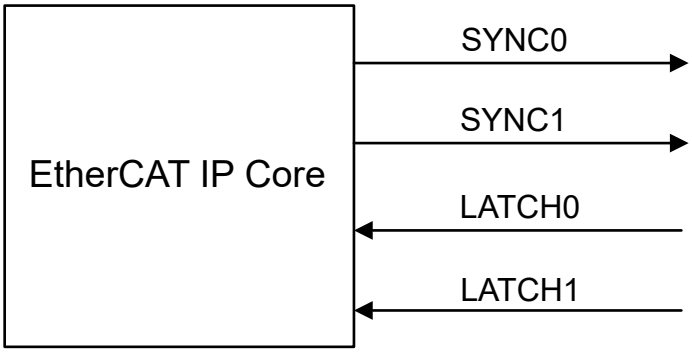


Figure 35-13. ESC SYNC and LATCH

This feature has sub-features as listed below. To enable the utility of these features at the application level, the related signals of DC are integrated tightly with the C2000 control loop logic, as well as, an external component that can be a possible implementation to trigger/latch other components on the board. This section explains the integration of these signals and related nuances of usage.

Distributed clock features:

- Clock Synchronization
- Sync Signal Generation
- Latch Signal event capture

Apart from usage for control loop synchronizations, these features are also used to synchronize the system and network time. These include the following:

- **System Time PDI Controlled:** Synchronize system time between two different EtherCAT networks.
- **Communication Timing:** To synchronize SubordinateDevice communication with sync signals, output or input events, and so on.

For further reading, including details on network time or network propagation delay, refer to the **ESC Hardware Data Sheet Section I - Technology (ethercat_esc_datasheet_sec1_technology)** document provided by Beckhoff/ETG at www.beckhoff.com.

35.2.10.1 Clock Synchronization

DC clock synchronization is the ability and procedure of the ESC to maintain the copy of the reference clock based on local clock (internal 64-bit time base) and other adjustment as derived by the procedure. The reference clock is the most accurate clock in the system and is typically held by one of the SubordinateDevices (Topologically first one after MainDevice is preferred). This time-base has a higher accuracy requirement and needs to periodically synchronize with an absolute time source like GPS or any other maintained time-base as in IEEE1588 network.

The synchronization order is as follows:

1. The MainDevice maintains a MainDevice clock that is either the absolute time synchronized or controls the reference clock.
2. The MainDevice queries the ESC time-base based on topology information and timestamps to determine the drift of the local clock for each ESC with regards to the reference clock.
3. The MainDevice programs the clock drift adjustments in each respective ESC.
4. With further periodic queries of the local clock, the MainDevice determines the drift of the local clock and keeps adjusting the clock so as to maintain the copies of the reference clocks on each ESCs in a tight limit.

While the ESC requires two clocks of 25MHz and 100MHz, the 100MHz clock is used for the internal time-base and supports the best accuracy possible. Refer to Section 35.2.6.2 for more details on clocking.

35.2.10.2 SYNC Signals

Sync signals are precise time controlled signals and are able to trigger time synchronized actions of the MainDevice. CPU core and RTDMA engine triggers are configured independently for every end point. The following sections describe ESC signal integration and various control triggers in the device.

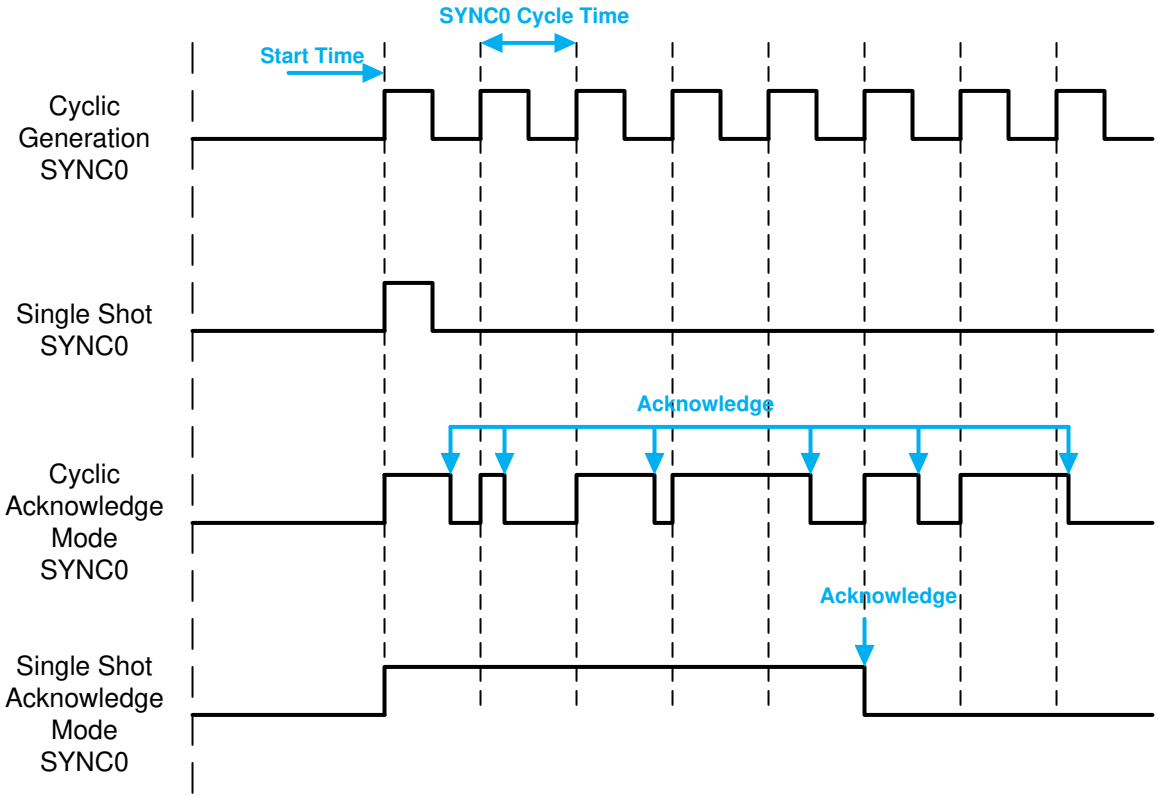


Figure 35-14. SYNC0 Signal Modes

- **SYNC0:** SYNC0 is the primary trigger and can be generated in either a cyclic (periodic event generation like rise edge at every x period) or one-shot mode. Furthermore, these modes can be either with or without acknowledge, such that when enabled with acknowledge mode, the next event is not generated until the acknowledgment is received from the controlling MainDevice. If the acknowledgment is delayed, the event is skipped and the next periodic event is generated. Remember that the acknowledgment can be part of the interrupt servicing from the PDI and such a delay in triggering the next event is acceptable since the servicing routine can accordingly take action for the period elapsed. These modes are shown in Figure 35-14.
- **SYNC1:** The SYNC1 generation follows the SYNC0 generation with a programmable delay and depending upon the delay time defined, SYNC1 can or cannot generate the pulse since the predefined delay from the SYNC0 event is newly measured only after the SYNC1 event (except for the start).

Note

- The SYNC0/1 when used in a system clock domain is stretched to at least 3 clock-wide pulse.
- When ESC goes through a reset, the SYNC outputs triggering external device events need to be kept at a safe value. At reset, the output values are 0 and this can be an active state when the reset occurs; therefore, the corresponding care of usage from the remote device must be taken.

35.2.10.2.1 Seeking Host Intervention

SYNC can be used to initiate the action from the local host or related RTDMA engine that can respond with control action or data transfers from the ESC IP.

These actions can be selectively initiated in terms of:

- Interrupts to CPU cores (CPU1, CLA, or CPU2)
- RTDMA requests to RTDMA cores (CPU1 RTDMA and CPU2 RTDMA)

Figure 35-15 depicts the SYNC connection diagram to the CPU subsystems.

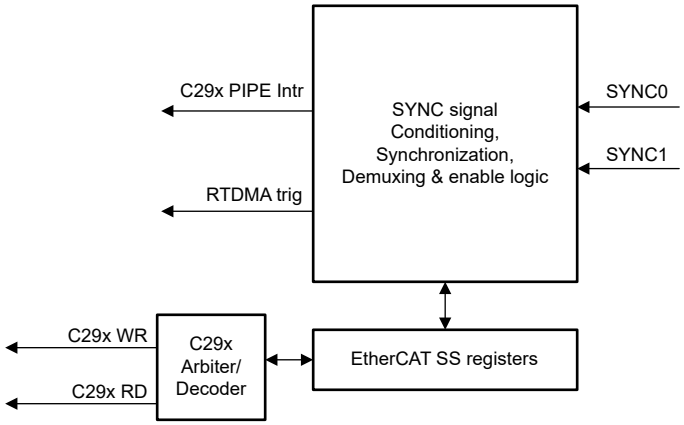


Figure 35-15. SYNC Integration for the HOST Intervention

Table 35-11 shows the Host Intervention selections, control and information that helps applications to route and handle the SYNC signals for respective Host intervention.

Table 35-11. ESC SYNC Integration Map

Destination	Source	Enable	Mask	Clear	Source Clock	Destination Clock	Destination Signaling
C29x Interrupt Controller Interrupt	SYNC0	ESCSS_SYNC0_ CONFIG[0]	ESCSS_INTR_ MASK[0]	ESCSS_INTR_ CLR[0]	ECAT.100MHz	C28x.SysClk	Pulse
C29x Interrupt Controller Interrupt	SYNC1	ESCSS_SYNC1_ CONFIG[0]	ESCSS_INTR_ MASK[1]	ESCSS_INTR_ CLR[1]	ECAT.100MHz	C28x.SysClk	Pulse
CLA Interrupt	SYNC0	ESCSS_SYNC0_ CONFIG[1]	NA	NA	ECAT.100MHz	C28x.SysClk	Pulse
CLA Interrupt	SYNC1	ESCSS_SYNC1_ CONFIG[1]	NA	NA	ECAT.100MHz	C28x.SysClk	Pulse
C29x RTDMA Trigger	SYNC0	ESCSS_SYNC0_ CONFIG[2]	NA	NA	ECAT.100MHz	C28x.SysClk	Pulse
C29x RTDMA Trigger	SYNC1	ESCSS_SYNC1_ CONFIG[2]	NA	NA	ECAT.100MHz	C28x.SysClk	Pulse
RTDMA Trigger	SYNC0	ESCSS_SYNC0_ CONFIG[4]	NA	NA	ECAT.100MHz	CPU2.SysClk	Pulse/Level
RTDMA Trigger	SYNC1	ESCSS_SYNC1_ CONFIG[4]	NA	NA	ECAT.100MHz	CPU2.SysClk	Pulse/Level

The difference between Enable and Mask is that Enable allows the conditioned and synchronized interrupt to be routed to the raw interrupt/trigger cause register, while Mask is a software control to allow raising an interrupt or not. Disabling the SYNC0 and SYNC1 on a respective trigger can loose any events that happen until SYNC0 and SYNC1 are enabled again.

Regarding the CLA:

- The CLA does not have access to EtherCAT, so the CLA does not have the MASK and Clear controls; however, the trigger starts the action on the CLA processing which, upon completion, must be acknowledged by CPU1 to clear the cause.

Regarding the RTDMA:

- The RTDMA triggers do not have Mask and Clear capabilities, as typically there is no feedback mechanism from the RTDMA to clear the trigger cause.
- Figure 35-16 is the symbolic view of RTDMA request source select. If a given source event triggers multiple events across different MainDevices, then the software has to make sure there is a status exchange before the cause is cleared.

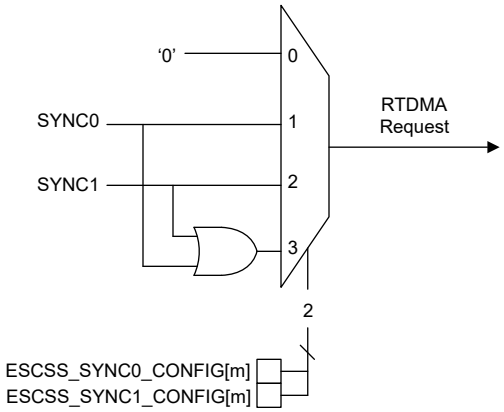


Figure 35-16. SYNC Event Muxing for Different Host RTDMA Triggers

35.2.10.3 LATCH Signals

Latch inputs to the ESC can be used to capture the time-stamp or register the GPI inputs. Two latch inputs are available. Latch enables external events:

- To take a system time snapshot of the EtherCAT network, measure time, or schedule further operations.
- To capture the status of the GPI inputs either as the status of connected external components or as part of control flow.

Both of the Latch inputs can be independently configured to operate on either the rising or falling edge of the signal. The LATCH events can be individually assigned either for the PDI control or EtherCAT controller control. Additionally, one shot or continuous mode is supported.

- In **one shot mode**, the next capture of the timestamp is done on a qualifying LATCH event only after the previous event is acknowledged.
- In **continuous mode**, timestamps are successively captured on a qualifying event, regardless of a preceding event being read or not.

Figure 35-17 depicts the different sources for the LATCH0/1 so as to fulfill the application requirements. These are explained for possible use.

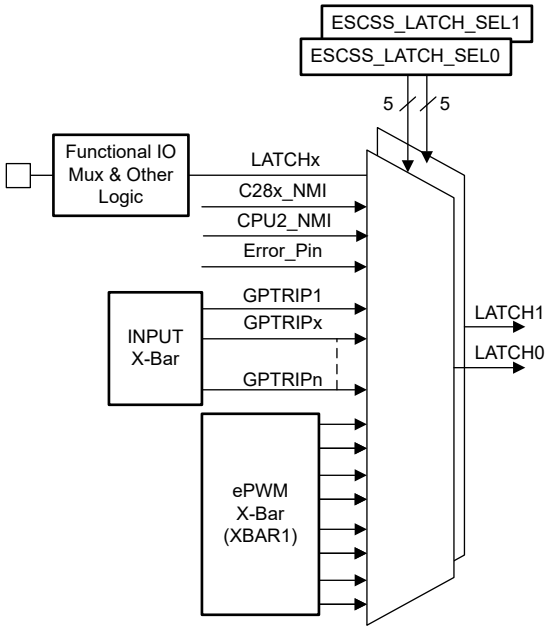


Figure 35-17. ESC Latch Input Integration

The details of connections and mux select to these muxes is shown in Table 35-12 and selection is possible individually for each of the LATCH0 or LATCH1 signal.

Table 35-12. ESC LATCH0/1 Trigger Table

ECAT_LATCH_SELx	Signal Hookup
0	ECAT_LATCH0
1	ECAT_LATCH1
2	CPU1NMI
3	CPU2NMI
4	ERRORSTS
5	INPUTXBAR1
6	INPUTXBAR2
7	INPUTXBAR3
8	INPUTXBAR4
9	INPUTXBAR5
10	INPUTXBAR6
11	INPUTXBAR7
12	INPUTXBAR8
13	INPUTXBAR9
14	INPUTXBAR10
15	INPUTXBAR11
16	INPUTXBAR12
17	INPUTXBAR13
18	INPUTXBAR14
19	INPUTXBAR15
20	INPUTXBAR16
21	EPWMXBAR1
22	EPWMXBAR2
23	EPWMXBAR3
24	EPWMXBAR4
25	EPWMXBAR5
26	EPWMXBAR6
27	EPWMXBAR7
28	EPWMXBAR8
29	CPU3NMI
30-31	Reserved

35.2.10.3.1 Timestamping

Timestamping allows the local application or remote EtherCAT MainDevice to measure the time events and plan the subsequent controls.

Timestamping: Device Internal Events

- Enable timestamping and logging of internal device events coming from control logic
- The same outputs of the EPWM crossbar that connect to the 8 EPWMs can be used as 8 input signals (PWMXBAROUT0-PWMXBAROUT7) for time stamping
- Note that there is no independent option for Latch muxing as far as EPWM cross-bar inputs are concerned; unless one of the TRIP outputs are not used in control loop and such a crossbar output is dedicated for LATCH0/1 toggle

Timestamping: External Events

- Enable external board components apart from the MCU to trigger timestamp capture
- For example, timestamp a periodic event or pulse/edge event from a senor when the sensor data is read or accessed
- Connecting the LATCH0/1 controls through the input crossbar allows timestamp capture based on any selected GPIO toggle (such as GPTRIP15 or GPTRIP16)
  - Two GPTRIPs are provided for independent GPIO choice for LATCH0 and LATCH1
  - GPTRIP1, 2, and 3 allow the same functionality; however, those can be utilized for EPWM trip zone functions where pins can or cannot need to match. In the case of a trip zone based on GPTRIP is to be latched, one of these inputs can be used.

Timestamping: Device Exception Events

- Log and timestamp exception events within the device (example: NMI exceptions) and periodically collect information to find out systemic issues in the system
- Can be used by local application and the remote MainDevice to diagnose or debug the system
- On CPU1, ERAD can be used to analyze particular accesses or data patterns/counts on the CPU bus. Refer to the *Embedded Real-time Analysis and Diagnostic (ERAD)* chapter for further information

35.2.10.4 Device Control and Synchronization

The primary advantage of DC functionality is when used in conjunction with device control functions. The precisely timed pulse/edge nature of the SYNC allows the ESC to synchronize the internal resources like PWMs, ECAPs (for capture as well as PWM) with remote system components. The following are the connections which are available for applications to program for allowing this synchronization.

35.2.10.4.1 Synchronization of PWM

The PWM Sync-chain on the MCU can be triggered either by the device external inputs from the GPIO via input cross-bar or the PWM internal events when the chain is programmed appropriately. The SYNC0/1 inputs in this implementation also act as the external sync inputs to the PWM sync-chain. The integration is shown in Figure 35-18.

The additional select register added to the trigger crossbar register allows independent programming of the EXTSYNCx from either GPTRIP or SYNC0/1.

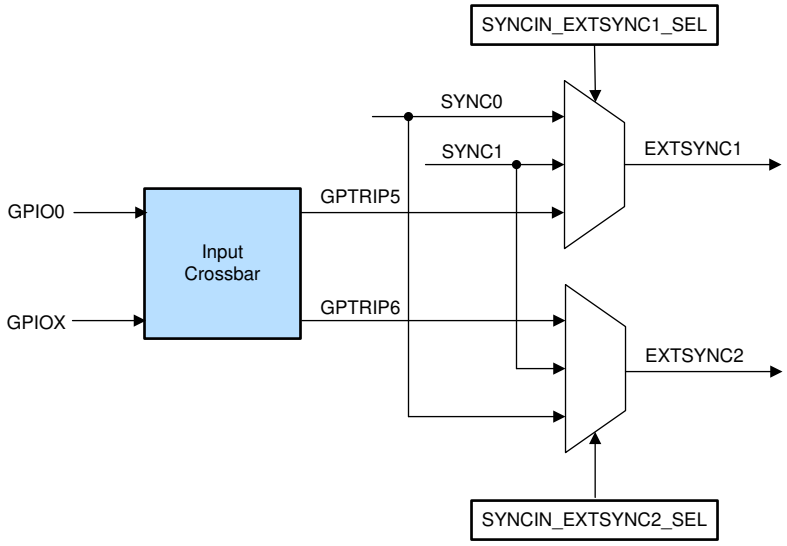


Figure 35-18. SYNC Integration for Control Functions - PWM SYNC

35.2.10.4.2 ECAP SYNC Inputs

ECAP supports accurate time capture of events that can be relatively checked against the series of events in vicinity and used for the control loop action. Additionally, ECAP has inbuilt capability of limited PWM action that can be triggered based on the selected input. Hence, the SYNC0/1 are also connected to ECAP input mux. The exact select value/vector for SYNC0/1 is defined in the *Enhanced Capture (eCAP)* chapter.

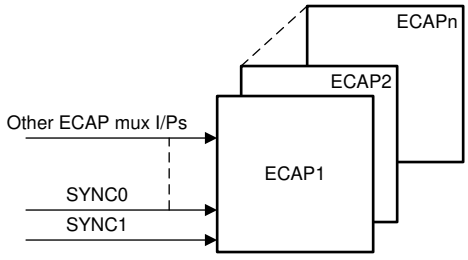


Figure 35-19. SYNC Integration for Control Functions – ECAP

35.2.10.4.3 SYNC Signal Conditioning and Rerouting

SYNC0/SYNC1 generation modes can generate various waveform patterns to create different trigger conditions. The SYNC0/1 signals are routed to the CLB, which can make conditional and delay based waveforms. Using the CLB, these signals can be processed for additional uses based on the application.

Additionally, this integration also allows routing SYNC0/1 signals with or without processing to other destinations within the device that are not explicitly connected. Details of the CLB input mux selects are explained in the *Configurable Logic Block (CLB)* chapter.

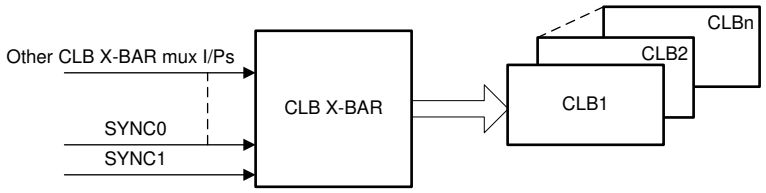


Figure 35-20. SYNC Integration for Signal Conditioning – CLB

35.3 Software Initialization Sequence and Allocating Ownership

This section details the software initialization sequence when configuring CPU1 or CPU2 as ESC owner. The CPU2 sequence includes details on allocating ownership of the ESC peripheral.

Table 35-13. CPU1 Software Initialization Sequence

Step	Action
1	General device initialization (configure clock, enable PLL, enable peripheral clocks except EtherCAT)
2	Configure Aux Clock for EtherCAT (if using Aux clock as source)
3	Configure GPIOs for EtherCAT (set pin configurations, set GPIO qualification mode, set pad configuration)
4	Initialize interrupts and register ISR handlers
5	Set EtherCAT clock source and divider. Then configure if EtherCAT PHY is clocked from device or external PHY clock.
6	Configure the EEPROM size
7	Bring ESC out of reset using system control register
8	Perform EtherCAT memory initialization and wait until memory initialization is complete
9	(Optional) Enable debug access to the EtherCAT registers
10	(Optional) Check that EEPROM loaded successfully
11	EtherCAT subsystem configurations for interrupt masking, SYNCx connections, and so on(1)

(1) Applications must make sure that ESC outputs are in a safe state until the EEPROM is loaded and that SYNC and LATCH are configured only after the EEPROM is loaded.

Table 35-14. CPU2 Software Initialization Sequence

Step	Core	Action
1	CPU1	General device initialization (configure clock, enable PLL, enable peripheral clocks except EtherCAT)
2	CPU1	Assign RAMs and Flash Banks to CPU2/CPU3
3	CPU1	Bring CPU2 out of reset using system control register
4	CPU1	Configure GPIOs for EtherCAT (set pin configurations, set GPIO qualification mode, set pad configuration)
5	CPU1	Configure Aux Clock for EtherCAT (if using Aux clock as source)
6	CPU1	Set EtherCAT source and clock divider. Then configure if EtherCAT PHY is clocked from device or external PHY clock
7	CPU1	Reset ESC using system control register
8	CPU1	Bring ESC out of reset using system control register
9	CPU1	Configure EtherCAT EEPROM Size
10	CPU1	Reset ESC using system control register
11	CPU1	Allocate ESC to CPU2/CPU3
12	CPU2/CPU3	Initialize Interrupts and Register ISR Handlers
13	CPU2/CPU3	Perform EtherCAT memory initialization and wait until memory initialization is complete
14	CPU2/CPU3	(Optional) Enable debug access to the EtherCAT registers
15	CPU2/CPU3	(Optional) Check that EEPROM Loaded successfully
16	CPU2/CPU3	EtherCAT subsytem configurations for interrupt masking, SYNCx connections and so on(1)

(1) Applications must make sure that ESC outputs are in a safe state until the EEPROM is loaded and that SYNC and LATCH are configured only after the EEPROM is loaded.

35.4 ESC Configuration Constants

The following is EtherCAT IP core configurations that are static by design.

Table 35-15. ESC Configuration Constants Table

Name	Value
ESC Type	0x91
Revision	0x0
Build	0x0
FMMU Supported	0x8
SyncManagers	0x8
RAM Size	0x10
Port Descriptor	0xF
ESC Features Supported	0x1CC
Product ID	0x0
Vendor ID	0x0

Table 35-16. ESC IP Register Constants Table

EtherCAT IP Register	Offset Address	Value(1)
PDI 8/16Bit asynchronous Microcontroller configuration	0x0150	0x00
Sync/Latch PDI configuration	0x0151	0x66
Pulse Length of SyncSignals	0x0982/0x0983	0x000A

(1) These values are not changed or overwritten by EEPROM contents.

35.5 Software

35.5.1 ECAT_SS Registers to Driverlib Functions

Table 35-17. ECAT_SS Registers to Driverlib Functions

File	Driverlib Function
ESCSS_IPREVNUM	
escss.h	ESCSS_readIPMinorRevNumber
escss.h	ESCSS_readIPMajorRevNumber
escss.h	ESCSS_readIPRevNumber
ESCSS_INTR_RIS	
escss.h	ESCSS_getRawInterruptStatus
escss.h	ESCSS_readRawInterruptStatus
ESCSS_INTR_MASK	
escss.h	ESCSS_setMaskedInterruptStatus
escss.h	ESCSS_resetMaskedInterruptStatus
ESCSS_INTR_MIS	
escss.h	ESCSS_getMaskedInterruptStatus
ESCSS_INTR_CLR	
escss.h	ESCSS_clearRawInterruptStatus
ESCSS_INTR_SET	
escss.c	ESCSS_setRawInterruptStatus
ESCSS_LATCH_SEL	
escss.h	ESCSS_selectLatch0Mux
escss.h	ESCSS_selectLatch1Mux
ESCSS_ACCESS_CTRL	
escss.h	ESCSS_configure16BitAsyncAccessWaitState
escss.h	ESCSS_enablePDIAccessTimeOut
escss.h	ESCSS_disablePDIAccessTimeOut
escss.h	ESCSS_enableDebugAccess
escss.h	ESCSS_disableDebugAccess
ESCSS_GPIN_DAT	
escss.h	ESCSS_readGPINData
escss.h	ESCSS_setGPINData
escss.h	ESCSS_resetGPINData
ESCSS_GPIN_PIPE	
escss.h	ESCSS_enableGPIPipelinedRegCaptureOnEvent
escss.h	ESCSS_disableGPIPipelinedRegCaptureOnEvent
ESCSS_GPIN_GRP_CAP_SEL	
escss.c	ESCSS_setGPINGroupCaptureTriggerSelect
ESCSS_GPOUT_DAT	
escss.h	ESCSS_readGPOUTData
ESCSS_GPOUT_PIPE	
escss.h	ESCSS_enableGPOUTPipelinedRegCaptureOnEvent
escss.h	ESCSS_disableGPOUTPipelinedRegCaptureOnEvent
ESCSS_GPOUT_GRP_CAP_SEL	
escss.c	ESCSS_setGPOUTGroupCaptureTriggerSelect
ESCSS_MEM_TEST	

Table 35-17. ECAT_SS Registers to Driverlib Functions (continued)

File	Driverlib Function
escss.h	ESCSS_initMemory
escss.h	ESCSS_getMemoryInitDoneStatusNonBlocking
escss.h	ESCSS_getMemoryInitDoneStatusBlocking
ESCSS_RESET_DEST_CONFIG	
escss.c	ESCSS_enableCPUReset
escss.c	ESCSS_disableCPUReset
escss.c	ESCSS_enableResetToNMI
escss.c	ESCSS_disableResetToNMI
escss.c	ESCSS_enableResetToInterrupt
escss.c	ESCSS_disableResetToInterrupt
ESCSS_SYNC0_CONFIG	
escss.c	ESCSS_configureSync0Connections
ESCSS_SYNC1_CONFIG	
escss.c	ESCSS_configureSync1Connections
ESCSS_CONFIG_LOCK	
escss.c	ESCSS_enableConfigurationLock
escss.c	ESCSS_enableIOConnectionLock
escss.c	ESCSS_disableIOConnectionLock
escss.h	ESCSS_isConfigurationLockEnabled
ESCSS_MISC_IO_CONFIG	
escss.c	ESCSS_enableResetInputFromGpioPad
escss.c	ESCSS_disableResetInputFromGpioPad
escss.c	ESCSS_enableESCEEPROMI2CIoPadConnection
escss.c	ESCSS_disableESCEEPROMI2CIoPadConnection
ESCSS_PHY_IO_CONFIG	
escss.c	ESCSS_configurePortCount
escss.c	ESCSS_enableAutoCompensationTxClkIOPad
escss.c	ESCSS_disableAutoCompensationTxClkIOPad
ESCSS_SYNC_IO_CONFIG	
escss.c	ESCSS_enableSync0GpioMuxConnection
escss.c	ESCSS_disableSync0GpioMuxConnection
escss.c	ESCSS_enableSync1GpioMuxConnection
escss.c	ESCSS_disableSync1GpioMuxConnection
ESCSS_LATCH_IO_CONFIG	
escss.c	ESCSS_enableLatch0GpioMuxConnection
escss.c	ESCSS_disableLatch0GpioMuxConnection
escss.c	ESCSS_enableLatch1GpioMuxConnection
escss.c	ESCSS_disableLatch1GpioMuxConnection
ESCSS_GPIN_SEL	
escss.h	ESCSS_enableGPIN
escss.h	ESCSS_disableGPIN
ESCSS_GPOUT_SEL	
escss.h	ESCSS_enableGPOUT
escss.h	ESCSS_disableGPOUT
ESCSS_LED_CONFIG	

Table 35-17. ECAT_SS Registers to Driverlib Functions (continued)

File	Driverlib Function
escss.h	ESCSS_enableLEDOptions
escss.h	ESCSS_disableLEDOptions
ESCSS_MISC_CONFIG	
escss.c	ESCSS_configureEEPROMSize
escss.h	ESCSS_configureTX0ShiftForTxEnaAndTxData
escss.h	ESCSS_configureTX1ShiftForTxEnaAndTxData
escss.h	ESCSS_enablePDIEmulation
escss.h	ESCSS_disablePDIEmulation
escss.h	ESCSS_configurePhyAddressOffset

35.5.2 ETHERNET Examples

NOTE: These examples are located in the C2000Ware installation at the following location:
C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/ethernet

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

35.6 ETHERCAT Registers

This Section describes the ETHERCAT Registers.

35.6.1 ETHERCAT Base Address Table

Table 35-18. ETHERCAT Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
ESCSS_REGS	ESC_SS_BASE	0x3038_8000	-	YES	YES	YES	YES	YES	-	YES
ESCSS_CONFIG_REGS	ESC_SS_CONFIG_BASE	0x3038_8200	-	YES	YES	YES	YES	YES	-	YES

35.6.2 ESCSS_REGS Registers

Table 35-19 lists the memory-mapped registers for the ESCSS_REGS registers. All register offset addresses not listed in Table 35-19 should be considered as reserved locations and the register contents should not be modified.

Table 35-19. ESCSS_REGS Registers

Offset	Acronym	Register Name	Protection
0h	ESCSS_IPREVNUM	IP Revision Number	
4h	ESCSS_INTR_RIS	EtherCATSS Interrupt Raw Status	
8h	ESCSS_INTR_MASK	EtherCATSS Interrupt Mask	
Ch	ESCSS_INTR_MIS	EtherCATSS Masked Interrupt Status	
10h	ESCSS_INTR_CLR	EtherCATSS Interrupt Clear	
14h	ESCSS_INTR_SET	EtherCATSS Interrupt Set to emulate	
18h	ESCSS_LATCH_SEL	Select for Latch0/1 inputs and LATCHIN input	
1Ch	ESCSS_ACCESS_CTRL	PDI interface access control config.	
20h	ESCSS_GPIN_DAT	GPIN data capture for debug & override	
24h	ESCSS_GPIN_PIPE	GPIN pipeline select	
28h	ESCSS_GPIN_GRP_CAP_SEL	GPIN pipe group capture trigger	
2Ch	ESCSS_GPOUT_DAT	GPOUT data capture for debug & override	
30h	ESCSS_GPOUT_PIPE	GPOUT pipeline select	
34h	ESCSS_GPOUT_GRP_CAP_SEL	GPOUT pipe group capture trigger	
38h	ESCSS_MEM_TEST	Memory Test Control	
3Ch	ESCSS_RESET_DEST_CONFIG	ResetOut impact or destination config	LOCK
40h	ESCSS_SYNC0_CONFIG	SYNC0 Configuration for various triggers	LOCK
44h	ESCSS_SYNC1_CONFIG	SYNC1 Configuration for various triggers	LOCK

Complex bit access types are encoded to fit into small table cells. Table 35-20 shows the codes that are used for access types in this section.

Table 35-20. ESCSS_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

35.6.2.1 ESCSS_IPREVNUM Register (Offset = 0h) [Reset = 00000000h]

ESCSS_IPREVNUM is shown in Figure 35-21 and described in Table 35-21.

Return to the Summary Table.

IP Revision number showing the Major & Minor IP versions 4 bit each

Figure 35-21. ESCSS_IPREVNUM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IP_REV_MAJOR				IP_REV_MINOR			
R-0-0h								R-0h				R-0h			

Table 35-21. ESCSS_IPREVNUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-4	IP_REV_MAJOR	R	0h	Major IP Type increment is hardcoded reset value which increments to signify major change in IP behavior in terms of data/control flow or new feature addition. Reset type: ECAT.IPRSn
3-0	IP_REV_MINOR	R	0h	Reset value for this register is hardcoded and increments with minor changes to the IP those will not increment IP Type, but the bug fixes and changes impact behavior or software control than previous silicon version. Reset type: ECAT.IPRSn

35.6.2.2 ESCSS_INTR_RIS Register (Offset = 4h) [Reset = 00000000h]

ESCSS_INTR_RIS is shown in Figure 35-22 and described in Table 35-22.

Return to the Summary Table.

Registers the Raw Interrupt status of different interrupt triggers regardless of mask.

Figure 35-22. ESCSS_INTR_RIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		MASTER_RESET_RIS	TIMEOUT_ERR_RIS	DMA_DONE_RIS	IRQ_RIS	SYNC1_RIS	SYNC0_RIS
R-0-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 35-22. ESCSS_INTR_RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	MASTER_RESET_RIS	R	0h	Indicates Raw Status of the EtherCAT Master Reset event , until cleared by ESCSS_INTR_CLR 0: EtherCAT Master Reset Event did not happen since last IP reset or last clear of this bit and ECAT Master reset programmed to be Interrupt to local host. 1: EtherCAT Master Reset Event has occured. This status gets updated regardless of interrupt mask, it is registered on the EtherCATSS clock domain, sticky and remains active till cleared using ESCSS_INTR_CLR. If simultaneous clear & incoming event on same clock edge the incoming event registering has priority and clear does not have effect. Information of this event is lost if ECAT Master event is programmed to be reset IP. Reset type: ECAT.IPRSn
4	TIMEOUT_ERR_RIS	R	0h	Indicates Raw Status of the past event on PDI access timeout Error, until cleared by ESCSS_INTR_CLR 0: PDI Access Timeout Error Event did not happen since reset or last clear of this bit. 1: PDI Access Timeout Error Event has triggered the interrupt/DMA trigger. This status gets updated regardless of interrupt mask as long as Timeout is enabled, it is registered on the EtherCATSS clock domain, sticky and remains active till cleared using ESCSS_INTR_CLR. If simultaneous clear & incoming event on same clock edge the incoming event registering has priority and clear does not have effect. Reset type: ECAT.IPRSn

Table 35-22. ESCSS_INTR_RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	DMA_DONE_RIS	R	0h	Indicates Raw Status of the past event on DMA Done, until cleared by ESCSS_INTR_CLR 0: DMA Done Event did not happen since reset or last clear of this bit. 1: DMA Done Event has triggered the interrupt/DMA trigger. This status gets updated regardless of interrupt mask, it is registered on the EtherCATSS clock domain, sticky and remains active till cleared using ESCSS_INTR_CLR. If simultaneous clear & incoming event on same clock edge the incoming event registering has priority and clear does not have effect. Reset type: ECAT.IPRSn
2	IRQ_RIS	R	0h	Indicates Raw Status of the past event on EtherCATSS IRQ, until cleared by ESCSS_INTR_CLR 0: EtherCATSS IRQ Event did not happen since reset or last clear of this bit. 1: EtherCATSS IRQ Event has triggered the interrupt/DMA trigger. This status gets updated regardless of interrupt mask, it is registered on the EtherCATSS clock domain, sticky and remains active till cleared using ESCSS_INTR_CLR. If simultaneous clear & incoming event on same clock edge the incoming event registering has priority and clear does not have effect. Reset type: ECAT.IPRSn
1	SYNC1_RIS	R	0h	Indicates Raw Status of the past event on SYNC1, until cleared by ESCSS_INTR_CLR 0: SYNC1 Event did not happen since reset or last clear of this bit. 1: SYNC1 Event has triggered the interrupt/DMA trigger. This status gets updated regardless of interrupt mask, it is registered on the EtherCATSS clock domain, sticky and remains active till cleared using ESCSS_INTR_CLR. If simultaneous clear & incoming event on same clock edge the incoming event registering has priority and clear does not have effect. Reset type: ECAT.IPRSn
0	SYNC0_RIS	R	0h	Indicates Raw Status of the past event on SYNC0, until cleared by ESCSS_INTR_CLR 0: SYNC0 Event did not happen since reset or last clear of this bit. 1: SYNC0 Event has triggered the interrupt/DMA trigger. This status gets updated regardless of interrupt mask, it is registered on the EtherCATSS clock domain, sticky and remains active till cleared using ESCSS_INTR_CLR. If simultaneous clear & incoming event on same clock edge the incoming event registering has priority and clear does not have effect. Reset type: ECAT.IPRSn

35.6.2.3 ESCSS_INTR_MASK Register (Offset = 8h) [Reset = 00000000h]

ESCSS_INTR_MASK is shown in Figure 35-23 and described in Table 35-23.

Return to the Summary Table.

Allows to mask individual interrupt cause impacting the interrupt

Figure 35-23. ESCSS_INTR_MASK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		MASTER_RESET_MASK	TIMEOUT_ERR_MASK	DMA_DONE_MASK	IRQ_MASK	SYNC1_MASK	SYNC0_MASK
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 35-23. ESCSS_INTR_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	MASTER_RESET_MASK	R/W	0h	Masks EtherCAT Master reset event against any effect on interrupts or other CPU Interrupts. 1: EtherCAT Master Reset affects the interrupt/DMA trigger. 0: EtherCAT Master Reset masked and does not affect Interrupt. Reset type: ECAT.IPRSn
4	TIMEOUT_ERR_MASK	R/W	0h	Masks PDI access timeout Error to have effect on interrupts or other CPU Interrupts. 1: PDI Access Timeout Errors affects the interrupt/DMA trigger. 0: PDI Access Timeout Errors masked and does not affect Interrupt. Reset type: ECAT.IPRSn
3	DMA_DONE_MASK	R/W	0h	Masks DMA Done status update to have effect on interrupts or other CPU Interrupts. 1: DMA Done affects the interrupt/DMA trigger. 0: DMA Done masked and does not affect Interrupt. Raw DMA Done status is updated regardless of this setting. Reset type: ECAT.IPRSn
2	IRQ_MASK	R/W	0h	Masks EtherCATSS IRQ to have effect on interrupts or other CPU/DMA triggers. 1: EtherCATSS IRQ affects the interrupt/DMA trigger. 0: EtherCATSS IRQ masked and does not affect Interrupt/DMA trigger. Raw EtherCATSS IRQ status is updated regardless of this setting. Reset type: ECAT.IPRSn
1	SYNC1_MASK	R/W	0h	Masks SYNC1 to have effect on interrupts or other CPU/DMA triggers as programmed in HOST_TRIG_MAP registers. 1: SYNC1 affects the interrupt/DMA trigger. 0: SYNC1 masked and does not affect Interrupt/DMA trigger. Raw SYNC1 status is updated regardless of this setting. Reset type: ECAT.IPRSn

Table 35-23. ESCSS_INTR_MASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SYNC0_MASK	R/W	0h	Masks SYNC0 to have effect on interrupts or other CPU/DMA triggers as programmed in HOST_TRIG_MAP registers. 1: SYNC0 affects the interrupt/DMA trigger. 0: SYNC0 masked and does not affect Interrupt/DMA trigger. Raw SYNC0 status is updated regardless of this setting. Reset type: ECAT.IPRSn

35.6.2.4 ESCSS_INTR_MIS Register (Offset = Ch) [Reset = 00000000h]

ESCSS_INTR_MIS is shown in Figure 35-24 and described in Table 35-24.

Return to the Summary Table.

Registers the Msked Interrupt status of different interrupt triggers. This is AND of RIS & MASK of respective fields

Figure 35-24. ESCSS_INTR_MIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		MASTER_RESET_MIS	TIMEOUT_ERR_MIS	DMA_DONE_MIS	IRQ_MIS	SYNC1_MIS	SYNC0_MIS
R-0-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 35-24. ESCSS_INTR_MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	MASTER_RESET_MIS	R	0h	Indicates Masked Interrupt status of the past event on EtherCAT Master Reset, until RIS cleared by ESCSS_INTR_CLR or by Reset. 0: No pending EtherCAT Master Reset interrupt, if configured and unmasked. 1: EtherCAT Master Reset Event has triggered the interrupt/DMA trigger and it's pending. This is MASK qualified RIS such that application can select to service or suppress the interrupt cause behind this. Upon reset or upon RIS clear through ESCSS_INTR_CLR this field is cleared. Reset type: ECAT.IPRSn
4	TIMEOUT_ERR_MIS	R	0h	Indicates Masked Interrupt status of the past event on PDI Access Timeout Error, until RIS cleared by ESCSS_INTR_CLR 0: No pending PDI Access Timeout Error interrupt. 1: PDI Access Timeout Error Event has triggered the interrupt/DMA trigger and it's pending. This is MASK qualified RIS such that application can select to service or suppress the interrupt cause behind this. Upon reset or upon RIS clear through ESCSS_INTR_CLR this field is cleared. Reset type: ECAT.IPRSn
3	DMA_DONE_MIS	R	0h	Indicates Masked Interrupt status of the past event on DMA Done, until RIS is cleared by ESCSS_INTR_CLR 0: No pending DMA Done interrupt. 1: DMA Done Event has triggered the interrupt/DMA trigger and it's pending. This is MASK qualified RIS such that application can select to service or suppress the interrupt cause behind this. Upon reset or upon RIS clear through ESCSS_INTR_CLR this field is cleared. Reset type: ECAT.IPRSn

Table 35-24. ESCSS_INTR_MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	IRQ_MIS	R	0h	Indicates Masked Interrupt status of the past event on EtherCATSS IRQ, until RIS is cleared by ESCSS_INTR_CLR 0: No pending EtherCATSS IRQ interrupt. 1: EtherCATSS IRQ Event has triggered the interrupt and it's pending. This is MASK qualified RIS such that application can select to service or suppress the interrupt cause behind this. Upon reset or upon RIS clear through ESCSS_INTR_CLR this field is cleared. Reset type: ECAT.IPRSn
1	SYNC1_MIS	R	0h	Indicates Masked Interrupt status of the past event on SYNC0, until RIS is cleared by ESCSS_INTR_CLR 0: No pending SYNC1 interrupt. 1: SYNC1 Event has triggered the interrupt/DMA trigger and it's pending. This is MASK qualified RIS such that application can select to service or suppress the interrupt cause behind this. Upon reset or upon RIS clear through ESCSS_INTR_CLR this field is cleared. Reset type: ECAT.IPRSn
0	SYNC0_MIS	R	0h	Indicates Masked Interrupt status of the past event on SYNC0, until RIS is cleared by ESCSS_INTR_CLR 0: No pending SYNC0 interrupt. 1: SYNC0 Event has triggered the interrupt/DMA trigger and it's pending. This is MASK qualified RIS such that application can select to service or suppress the interrupt cause behind this. Upon reset or upon RIS clear through ESCSS_INTR_CLR this field is cleared. Reset type: ECAT.IPRSn

35.6.2.5 ESCSS_INTR_CLR Register (Offset = 10h) [Reset = 00000000h]

ESCSS_INTR_CLR is shown in Figure 35-25 and described in Table 35-25.

Return to the Summary Table.

Individual Interrupt cause clear register

Figure 35-25. ESCSS_INTR_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		MASTER_RESET_CLR	TIMEOUT_ERR_CLR	DMA_DONE_CLR	IRQ_CLR	SYNC1_CLR	SYNC0_CLR
R-0-0h		R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 35-25. ESCSS_INTR_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	MASTER_RESET_CLR	R-0/W1C	0h	Clears EtherCAT Master Reset raw Status (Common Mask & Clear among all hosts, only one host expected accessing). Write 1 clears the Raw status of the EtherCAT Master reset, read always returns 0. Reset type: ECAT.IPRSn
4	TIMEOUT_ERR_CLR	R-0/W1C	0h	Clears PDI access timeout Error raw Status (Common Mask & Clear among all hosts, only one host expected accessing). Write 1 clears the Raw status of the PDI Access Timeout Error, read always returns 0. Reset type: ECAT.IPRSn
3	DMA_DONE_CLR	R-0/W1C	0h	Clears DMA Done raw Status (Common Mask & Clear among all hosts, only one host expected accessing). Write 1 clears the Raw status of the DMA Done, read always returns 0. Reset type: ECAT.IPRSn
2	IRQ_CLR	R-0/W1C	0h	Clears EtherCATSS IRQ raw Status (Common Mask & Clear among all hosts, only one host expected accessing). Write 1 clears the Raw status of the EtherCATSS IRQ, read always returns 0. Reset type: ECAT.IPRSn
1	SYNC1_CLR	R-0/W1C	0h	Clears SYNC1 raw Status (Common Mask & Clear among all hosts, only one host expected accessing). Write 1 clears the Raw status of the SYNC1, read always returns 0. Reset type: ECAT.IPRSn
0	SYNC0_CLR	R-0/W1C	0h	Clears SYNC0 raw Status (Common Mask & Clear among all hosts, only one host expected accessing). Write 1 clears the Raw status of the SYNC0, read always returns 0. Reset type: ECAT.IPRSn

35.6.2.6 ESCSS_INTR_SET Register (Offset = 14h) [Reset = 00000000h]

ESCSS_INTR_SET is shown in Figure 35-26 and described in Table 35-26.

Return to the Summary Table.

Individual Interrupt cause set register to emulate the interrupt cause

Figure 35-26. ESCSS_INTR_SET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
WRITE_KEY							
R-0/W-0h							
7	6	5	4	3	2	1	0
RESERVED		MASTER_RESET_SET	TIMEOUT_ERR_SET	DMA_DONE_SET	IRQ_SET	SYNC1_SET	SYNC0_SET
R-0-0h		R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 35-26. ESCSS_INTR_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-8	WRITE_KEY	R-0/W	0h	The key value should be 0xa5 for the writes to Set bits to take effect. Writes of other values will be ignored. Reset type: ECAT.IPRSn
7-6	RESERVED	R-0	0h	Reserved
5	MASTER_RESET_SET	R-0/W1S	0h	Sets EtherCAT Master Reset raw Status (Common Mask & Clear among all hosts, only one host expected accessing). Write 1 Sets the Raw status of the PDI Access Timeout Error, read always returns 0. Note this emulation can only assert interrupt to CPU but it can not reset the EtherCAT IP. Reset type: ECAT.IPRSn
4	TIMEOUT_ERR_SET	R-0/W1S	0h	Sets PDI access timeout Error raw Status (Common Mask & Clear among all hosts, only one host expected accessing). Write 1 Sets the Raw status of the PDI Access Timeout Error, read always returns 0. Reset type: ECAT.IPRSn
3	DMA_DONE_SET	R-0/W1S	0h	Sets DMA Done raw Status (Common Mask & Clear among all hosts, only one host expected accessing). Write 1 Sets the Raw status of the DMA Done, read always returns 0. Reset type: ECAT.IPRSn
2	IRQ_SET	R-0/W1S	0h	Sets EtherCATSS IRQ raw Status (Common Mask & Clear among all hosts, only one host expected accessing). Write 1 sets the Raw status of the EtherCATSS IRQ, read always returns 0. Reset type: ECAT.IPRSn
1	SYNC1_SET	R-0/W1S	0h	Sets SYNC1 raw Status (Common Mask & Clear among all hosts, only one host expected accessing). Write 1 Sets the Raw status of the SYNC1, read always returns 0. Reset type: ECAT.IPRSn

Table 35-26. ESCSS_INTR_SET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SYNC0_SET	R-0/W1S	0h	Sets SYNC0 raw Status (Common Mask & Clear among all hosts, only one host expected accessing). Write 1 sets the Raw status of the SYNC0, read always returns 0. Reset type: ECAT.IPRSn

35.6.2.7 ESCSS_LATCH_SEL Register (Offset = 18h) [Reset = 00000000h]

ESCSS_LATCH_SEL is shown in Figure 35-27 and described in Table 35-27.

Return to the Summary Table.

Select for LATCH0/1 input Triggers as well as LATCHIN used for registering the GPIs.

Figure 35-27. ESCSS_LATCH_SEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							RESERVED
R-0-0h							R/W-0h
15	14	13	12	11	10	9	8
RESERVED			LATCH1_SELECT				
R-0-0h			R/W-0h				
7	6	5	4	3	2	1	0
RESERVED			LATCH0_SELECT				
R-0-0h			R/W-0h				

Table 35-27. ESCSS_LATCH_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R-0	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15-13	RESERVED	R-0	0h	Reserved
12-8	LATCH1_SELECT	R/W	0h	Mux Select for LATCH1 input to ECATSS. Refer device specification for details of mux select options. Reset type: ECAT.IPRSn
7-5	RESERVED	R-0	0h	Reserved
4-0	LATCH0_SELECT	R/W	0h	Mux Select for LATCH0 input to ECATSS. Refer device specification for details of mux select options. Reset type: ECAT.IPRSn

35.6.2.8 ESCSS_ACCESS_CTRL Register (Offset = 1Ch) [Reset = 00000400h]

ESCSS_ACCESS_CTRL is shown in Figure 35-28 and described in Table 35-28.

Return to the Summary Table.

Wait state control for EtherCAT access

Figure 35-28. ESCSS_ACCESS_CTRL Register

31	30	29	28	27	26	25	24
RESERVED				TIMEOUT_COUNT			
R-0-0h				R/W-0h			
23	22	21	20	19	18	17	16
TIMEOUT_COUNT							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED					ENABLE_PARALLEL_PORT_ACCESS	ENABLE_DEBUG_ACCESS	RESERVED
R-0-0h					R/W-1h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
EN_TIMEOUT	WAIT_STATES						
R/W-0h	R/W-0h						

Table 35-28. ESCSS_ACCESS_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R-0	0h	Reserved
27-16	TIMEOUT_COUNT	R/W	0h	This is the cycle count in SYSCLK cycles after which an access on the EtherCAT Async interface will be aborted and CPU will be given back a READY. Reset type: ECAT.IPRSn
15-11	RESERVED	R-0	0h	Reserved
10	ENABLE_PARALLEL_PORT_ACCESS	R/W	1h	Enabled memory accesses through the parallel port interface. 0: Memory accesses using parallel port are not allowed to go through and will result in an error from the peripheral bridge to the ESM. 1: Memory accesses using parallel port are allowed through the Bridge. Reset type: ECAT.IPRSn
9	ENABLE_DEBUG_ACCESS	R/W	0h	Enabled debug accesses through the PDI interface. 0: Debug accesses are not allowed to go through. 1: Debug accesses are allowed through the Bridge. Bridge logic will ensure that access will not hang. Reset type: ECAT.IPRSn
8	RESERVED	R/W	0h	Reserved
7	EN_TIMEOUT	R/W	0h	Enables the Timeout features which counts programmed number of Sys clocks before the Local host aborts the transaction. 0: Timeout feature is not enabled on PDI interface. 1: The timeout counter starts counting upon BUSY is asserted by EtherCAT IP. Reset type: ECAT.IPRSn
6-0	WAIT_STATES	R/W	0h	This is the predefined minimum number of wait-states which the VBUS bridge will put out accesses on the 16-bit Async interface. Reset type: ECAT.IPRSn

35.6.2.9 ESCSS_GPIN_DAT Register (Offset = 20h) [Reset = 00000000h]

ESCSS_GPIN_DAT is shown in Figure 35-29 and described in Table 35-29.

Return to the Summary Table.

GPI data status for debug & overridwe

Figure 35-29. ESCSS_GPIN_DAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIN_DAT																															
R/W-0h																															

Table 35-29. ESCSS_GPIN_DAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPIN_DAT	R/W	0h	Local GPIN data register connects to GPIN pipelined register for debug & override purposes. Note: The copy of this register readable by the CPU is provided without synchronization, therefore multiple reads should be performed to confirm a stable value before using it. Reset type: ECAT.IPRSn

35.6.2.10 ESCSS_GPIN_PIPE Register (Offset = 24h) [Reset = 00000000h]

ESCSS_GPIN_PIPE is shown in Figure 35-30 and described in Table 35-30.

Return to the Summary Table.

Register to select raw PAD input or pipelined input be presented to ESC.

Figure 35-30. ESCSS_GPIN_PIPE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPI_PIPE																															
R/W-0h																															

Table 35-30. ESCSS_GPIN_PIPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPI_PIPE	R/W	0h	Enables the connection of GPIN to EtherCATSS through pipelined register as against the direct from IO. 0: The connection is directly from the IO pad 1: Connection is through the pipelined register which is captured on programmed event. Reset type: ECAT.IPRSn

35.6.2.11 ESCSS_GPIN_GRP_CAP_SEL Register (Offset = 28h) [Reset = 00000000h]

ESCSS_GPIN_GRP_CAP_SEL is shown in Figure 35-31 and described in Table 35-31.

Return to the Summary Table.

Register to configure trigger select for the group of 8 IOs together.

Figure 35-31. ESCSS_GPIN_GRP_CAP_SEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED	GPI_GRP_CAP_SEL3			RESERVED	GPI_GRP_CAP_SEL2		
R-0-0h	R/W-0h			R-0-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	GPI_GRP_CAP_SEL1			RESERVED	GPI_GRP_CAP_SEL0		
R-0-0h	R/W-0h			R-0-0h	R/W-0h		

Table 35-31. ESCSS_GPIN_GRP_CAP_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R-0	0h	Reserved
14-12	GPI_GRP_CAP_SEL3	R/W	0h	Selects the trigger to capture the IO input in pipeline register for GPI31-24 0: Start Of Frame as capture trigger 1-3: Reserved, selects default value of 0. 4: SYNC0 as capture trigger 5: SYNC1 as capture trigger 6: LATCH0 as capture trigger 7: LATCH1 as capture trigger Reset type: ECAT.IPRSn
11	RESERVED	R-0	0h	Reserved
10-8	GPI_GRP_CAP_SEL2	R/W	0h	Selects the trigger to capture the IO input in pipeline register for GPI23-16 0: Start Of Frame as capture trigger 1-3: Reserved, selects default value of 0. 4: SYNC0 as capture trigger 5: SYNC1 as capture trigger 6: LATCH0 as capture trigger 7: LATCH1 as capture trigger Reset type: ECAT.IPRSn
7	RESERVED	R-0	0h	Reserved
6-4	GPI_GRP_CAP_SEL1	R/W	0h	Selects the trigger to capture the IO input in pipeline register for GPI15-8 0: Start Of Frame as capture trigger 1-3: Reserved, selects default value of 0. 4: SYNC0 as capture trigger 5: SYNC1 as capture trigger 6: LATCH0 as capture trigger 7: LATCH1 as capture trigger Reset type: ECAT.IPRSn
3	RESERVED	R-0	0h	Reserved

Table 35-31. ESCSS_GPIN_GRP_CAP_SEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	GPI_GRP_CAP_SEL0	R/W	0h	Selects the trigger to capture the IO input in pipeline register for GPI7-0 0: Start Of Frame as capture trigger 1-3: Reserved, selects default value of 0. 4: SYNC0 as capture trigger 5: SYNC1 as capture trigger 6: LATCH0 as capture trigger 7: LATCH1 as capture trigger Reset type: ECAT.IPRSn

35.6.2.12 ESCSS_GPOUT_DAT Register (Offset = 2Ch) [Reset = 00000000h]

ESCSS_GPOUT_DAT is shown in Figure 35-32 and described in Table 35-32.

Return to the Summary Table.

GPO data capture for debug & overridwe

Figure 35-32. ESCSS_GPOUT_DAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPOUT_DAT																															
R-0h																															

Table 35-32. ESCSS_GPOUT_DAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPOUT_DAT	R	0h	Local GPOUT data register which is synchronised version on SysClk, each bit is represents GPO IO Read is allowed for CPU to process (IO extender or so if required). Reset type: ECAT.IPRSn

35.6.2.13 ESCSS_GPOUT_PIPE Register (Offset = 30h) [Reset = 00000000h]

ESCSS_GPOUT_PIPE is shown in Figure 35-33 and described in Table 35-33.

Return to the Summary Table.

Register to select pipeline of ESC output against direct route to IO pad on per IO based.

Figure 35-33. ESCSS_GPOUT_PIPE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPO_PIPE																															
R/W-0h																															

Table 35-33. ESCSS_GPOUT_PIPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPO_PIPE	R/W	0h	Enables the connection of EtherCATSS GPO output to the IO pad through pipelined register as against the direct connection. 0: The connection is directly to the IO pad 1: Connection is through the pipelined register which captures EtherCATSS o/p on programmed event. Reset type: ECAT.IPRSn

35.6.2.14 ESCSS_GPOUT_GRP_CAP_SEL Register (Offset = 34h) [Reset = 00000000h]

ESCSS_GPOUT_GRP_CAP_SEL is shown in Figure 35-34 and described in Table 35-34.

Return to the Summary Table.

Register to configure trigger select for pipelined register in group of 8 IOs together.

Figure 35-34. ESCSS_GPOUT_GRP_CAP_SEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED		GPO_GRP_CAP_SEL3		RESERVED		GPO_GRP_CAP_SEL2	
R-0-0h		R/W-0h		R-0-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED		GPO_GRP_CAP_SEL1		RESERVED		GPO_GRP_CAP_SEL0	
R-0-0h		R/W-0h		R-0-0h		R/W-0h	

Table 35-34. ESCSS_GPOUT_GRP_CAP_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	Reserved
13-12	GPO_GRP_CAP_SEL3	R/W	0h	Selects the trigger to capture the EtherCATSS output in pipeline register for GPO31-24 0: End Of Frame as capture trigger 1: SYNC0 as capture trigger 2: SYNC1 as capture trigger 3: WDTrig as capture trigger Reset type: ECAT.IPRSn
11-10	RESERVED	R-0	0h	Reserved
9-8	GPO_GRP_CAP_SEL2	R/W	0h	Selects the trigger to capture the EtherCATSS output in pipeline register for GPO23-16 0: End Of Frame as capture trigger 1: SYNC0 as capture trigger 2: SYNC1 as capture trigger 3: WDTrig as capture trigger Reset type: ECAT.IPRSn
7-6	RESERVED	R-0	0h	Reserved
5-4	GPO_GRP_CAP_SEL1	R/W	0h	Selects the trigger to capture the EtherCATSS output in pipeline register for GPO15-8 0: End Of Frame as capture trigger 1: SYNC0 as capture trigger 2: SYNC1 as capture trigger 3: WDTrig as capture trigger Reset type: ECAT.IPRSn
3-2	RESERVED	R-0	0h	Reserved

Table 35-34. ESCSS_GPOUT_GRP_CAP_SEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPO_GRP_CAP_SEL0	R/W	0h	Selects the trigger to capture the EtherCATSS output in pipeline register for GPO7-0 0: End Of Frame as capture trigger 1: SYNC0 as capture trigger 2: SYNC1 as capture trigger 3: WDTrig as capture trigger Reset type: ECAT.IPRSn

35.6.2.15 ESCSS_MEM_TEST Register (Offset = 38h) [Reset = 00000000h]

ESCSS_MEM_TEST is shown in Figure 35-35 and described in Table 35-35.

Return to the Summary Table.

This register controls access to memory test mode

Figure 35-35. ESCSS_MEM_TEST Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MEM_INIT_DONE	INITIATE_MEM_INIT
R-0-0h						R-0h	R-0/W1S-0h

Table 35-35. ESCSS_MEM_TEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	MEM_INIT_DONE	R	0h	Read-only status bit indicating memory initialisation completion. Gets self cleared with INITIATE_MEM_INIT is written '1'. Reset type: ECAT.IPRSn
0	INITIATE_MEM_INIT	R-0/W1S	0h	Memory Initialisation Trigger When set 1 the memory wrapper starts initialisation of DPRAM including parity programming. The bit gets Autocleared after memory initialisation starts. Write of 0 has no effect. Reset type: ECAT.IPRSn

35.6.2.16 ESCSS_RESET_DEST_CONFIG Register (Offset = 3Ch) [Reset = 00000000h]

ESCSS_RESET_DEST_CONFIG is shown in Figure 35-36 and described in Table 35-36.

Return to the Summary Table.

EtherCAT RESET_OUT configuration

Figure 35-36. ESCSS_RESET_DEST_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
WRITE_KEY							
R-0/W-0h							
7	6	5	4	3	2	1	0
DEVICE_RESET_EN	RESERVED				CPU_INT_EN	CPU_NMI_EN	CPU_RESET_EN
R/W-0h	R-0-0h				R/W-0h	R/W-0h	R/W-0h

Table 35-36. ESCSS_RESET_DEST_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-8	WRITE_KEY	R-0/W	0h	The key value should be 0xa5 for the writes to this register to take effect. Writes of other values will be ignored. Reset type: ECAT.IPRSn
7	DEVICE_RESET_EN	R/W	0h	Enable the EtherCAT RESET_OUT which is combination of IP Reset out and Pin reset to drive the Device Reset (XRSn) 0: EtherCAT RESET_OUT only drives the EtherCATSS & companion component reset to PHY 1: EtherCAT RESET_OUT drives EtherCATSS, external PHY reset and device by connecting this net on to device XRSn User's note: The connection from IP Resetout to EtherCAT RESET_OUT has no relation with this selection. Reset type: ECAT.XRSn
6-3	RESERVED	R-0	0h	Reserved
2	CPU_INT_EN	R/W	0h	Enable for resetout to drive the interrupt to CPU which it belongs to. 0: IP Resetout does not drive the CPU interrupt. 1: IP Resetout drives interrupt to the CPU master to which EtherCATSS belongs. The Host completes the reset through System control Soft reset after completing required context save or tasks if any. Reset type: ECAT.IPRSn
1	CPU_NMI_EN	R/W	0h	Enable for resetout to drive the CPU NMI 0: IP Resetout does not drive the CPU NMI. 1: IP Resetout drives CPU NMI to which it belongs. NMI handler is expected to complete the required taks or context save if any and then reset the EtherCAT through the system control soft reset. Reset type: ECAT.IPRSn

Table 35-36. ESCSS_RESET_DEST_CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	CPU_RESET_EN	R/W	0h	Enables EtherCAT Reset to drive the IP & PHY reset 0: EtherCAT Reset does not drive reset connection. 1: EtherCAT Reset drives EtherCAT IP and PHY Reset EtherCAT Reset Combines Master Reset, PDI sequence Reset, RESET_IN, System control soft Reset This selection is to drive the Master & PDI Reset to this combination. When this bit is set 0, application shall configure NMI/Interrupt to eventually complete the reset through system control soft reset. Reset type: ECAT.XRSn

35.6.2.17 ESCSS_SYNC0_CONFIG Register (Offset = 40h) [Reset = 00000000h]

ESCSS_SYNC0_CONFIG is shown in Figure 35-37 and described in Table 35-37.

Return to the Summary Table.

SYNC0 Triggers enable for Host events like Interrupts, DMA triggers across all masters & GPIO

Figure 35-37. ESCSS_SYNC0_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
WRITE_KEY							
R-0/W-0h							
7	6	5	4	3	2	1	0
RESERVED			RESERVED	RESERVED	DMAx_TRIG_EN	RESERVED	CPUx_INT_EN
R-0-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 35-37. ESCSS_SYNC0_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-8	WRITE_KEY	R-0/W	0h	The key value should be 0xa5 for the writes to this register to take effect. Writes of other values will be ignored. Reset type: ECAT.IPRSn
7-5	RESERVED	R-0	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	DMAx_TRIG_EN	R/W	0h	Makes the connection from SYNC0 output to DMA Trigger. 0: SYNC0 does not contribute to DMA trigger. 1: SYNC0 toggle Triggers the DMA Transfer. Reset type: ECAT.IPRSn
1	RESERVED	R/W	0h	Reserved
0	CPUx_INT_EN	R/W	0h	Makes the connection from SYNC0 output to CPUx Interrupt. 0: SYNC0 does not contribute to CPUx Interrupt regardless of mask. 1: SYNC0 follows the PIE interrupt behavior as controlled by RIS,MASK,CLR register pairs. Reset type: ECAT.IPRSn

35.6.2.18 ESCSS_SYNC1_CONFIG Register (Offset = 44h) [Reset = 00000000h]

ESCSS_SYNC1_CONFIG is shown in Figure 35-38 and described in Table 35-38.

Return to the Summary Table.

SYNC1 Triggers enable for Host events like Interrupts, DMA triggers across all masters & GPIO

Figure 35-38. ESCSS_SYNC1_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
WRITE_KEY							
R-0/W-0h							
7	6	5	4	3	2	1	0
RESERVED			RESERVED	RESERVED	DMAx_TRIG_EN	RESERVED	CPUx_INT_EN
R-0-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 35-38. ESCSS_SYNC1_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-8	WRITE_KEY	R-0/W	0h	The key value should be 0xa5 for the writes to this register to take effect. Writes of other values will be ignored. Reset type: ECAT.IPRSn
7-5	RESERVED	R-0	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	DMAx_TRIG_EN	R/W	0h	Makes the connection from SYNC1 output to DMA Trigger. 0: SYNC1 does not contribute to DMA trigger. 1: SYNC1 toggle Triggers the DMA Transfer. Reset type: ECAT.IPRSn
1	RESERVED	R/W	0h	Reserved
0	CPUx_INT_EN	R/W	0h	Makes the connection from SYNC1 output to CPUx Interrupt. 0: SYNC1 does not contribute to CPUx Interrupt regardless of mask. 1: SYNC1 follows the interrupt behavior as controlled by RIS,MASK,CLR register pairs. Reset type: ECAT.IPRSn

35.6.3 ESCSS_CONFIG_REGS Registers

Table 35-39 lists the memory-mapped registers for the ESCSS_CONFIG_REGS registers. All register offset addresses not listed in Table 35-39 should be considered as reserved locations and the register contents should not be modified.

Table 35-39. ESCSS_CONFIG_REGS Registers

Offset	Acronym	Register Name	Protection
0h	ESCSS_CONFIG_LOCK	EtherCATSS Configuration Lock	
4h	ESCSS_MISC_IO_CONFIG	RESET_IN, EEPROM IO connections select	LOCK
8h	ESCSS_PHY_IO_CONFIG	Control Register of ESCSS	
Ch	ESCSS_SYNC_IO_CONFIG	SYNC Signals IO configurations	LOCK
10h	ESCSS_LATCH_IO_CONFIG	LATCH inputs IO pad select	LOCK
14h	ESCSS_GPIN_SEL	GPIN Select between IO PAD & tieoff	LOCK
1Ch	ESCSS_GPOUT_SEL	GPOUT IO pad connect select	LOCK
24h	ESCSS_LED_CONFIG	Selection of LED o/p connect to IO pad	LOCK
28h	ESCSS_MISC_CONFIG	Miscelleneous Configuration	LOCK

Complex bit access types are encoded to fit into small table cells. Table 35-40 shows the codes that are used for access types in this section.

Table 35-40. ESCSS_CONFIG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value

35.6.3.1 ESCSS_CONFIG_LOCK Register (Offset = 0h) [Reset = 00000000h]

ESCSS_CONFIG_LOCK is shown in Figure 35-39 and described in Table 35-41.

Return to the Summary Table.

Lock bit for EtherCAT configuration registers

Figure 35-39. ESCSS_CONFIG_LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
WRITE_KEY							
R-0/W-0h							
7	6	5	4	3	2	1	0
RESERVED			IO_CONFIG_ENABLE	RESERVED			LOCK_ENABLE
R-0-0h			R/W-0h	R-0-0h			R/WSonce-0h

Table 35-41. ESCSS_CONFIG_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-8	WRITE_KEY	R-0/W	0h	The key value should be 0xa5 for the writes to bit 0 take effect. Writes of other values will be ignored. Reset type: ECAT.XRSn
7-5	RESERVED	R-0	0h	Reserved
4	IO_CONFIG_ENABLE	R/W	0h	This bit enables the IO configurations allowing the EtherCAT ports to take effect. Till this bit is written EtherCAT ports are not connected to the IO pad. Enable takes effect when this bit is set to 1. Changing IO selections or IO configurations after this bit is set can have unpredictable IO behavior on the device IOs. Reset type: ECAT.XRSn
3-1	RESERVED	R-0	0h	Reserved
0	LOCK_ENABLE	R/WSonce	0h	This bit enables locking the contents of all the EtherCAT configuration registers. The lock takes effect when this bit is set to 1. This bit can be set only once after ecatXRSN and gets reset after the next ecatXRSN. Reset type: ECAT.XRSn

35.6.3.2 ESCSS_MISC_IO_CONFIG Register (Offset = 4h) [Reset = 00000002h]

ESCSS_MISC_IO_CONFIG is shown in Figure 35-40 and described in Table 35-42.

Return to the Summary Table.

Condiguration of RESET_IN, EEPROM I2C connections

Figure 35-40. ESCSS_MISC_IO_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
WRITE_KEY							
R-0/W-0h							
7	6	5	4	3	2	1	0
RESERVED						EEPROM_I2C_IO_EN	RESETIN_GPIO_EN
R-0-0h						R/W-1h	R/W-0h

Table 35-42. ESCSS_MISC_IO_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-8	WRITE_KEY	R-0/W	0h	The key value should be 0xa5 for the writes to this register to take effect. Writes of other values will be ignored. Reset type: ECAT.XRSn
7-2	RESERVED	R-0	0h	Reserved
1	EEPROM_I2C_IO_EN	R/W	1h	Enables connecting EtherCAT I2C connections to IOPAD for EEPROM control 0: EEPROM I2C Connections are not connected to IOPAD. 1: EEPROM I2C connections are driving the IOPAD connections. Reset type: ECAT.XRSn
0	RESETIN_GPIO_EN	R/W	0h	Acts as enabled to receive the Reset input from GPIO pad. 0: RESET_IN GPIO pad is not enabled, only SW & PMM resets affect EtherCAT reset 1: RESET_IN GPIO pad input is connected in reset input cone. Reset type: ECAT.XRSn

35.6.3.3 ESCSS_PHY_IO_CONFIG Register (Offset = 8h) [Reset = 00000044h]

ESCSS_PHY_IO_CONFIG is shown in Figure 35-41 and described in Table 35-43.

Return to the Summary Table.

PHY Type, clock source type select

Figure 35-41. ESCSS_PHY_IO_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
WRITE_KEY							
R-0/W-0h							
7	6	5	4	3	2	1	0
RESERVED	TX_CLK_AUTO_COMP	RESERVED		PHY_PORT_CNT		RESERVED	
R/W-0h	R/W-1h	R/W-0h		R/W-1h		R/W-0h	

Table 35-43. ESCSS_PHY_IO_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-8	WRITE_KEY	R-0/W	0h	The key value should be 0xa5 for the writes to bit 0 take effect. Writes of other values will be ignored. Reset type: ECAT.XRSn
7	RESERVED	R/W	0h	Reserved
6	TX_CLK_AUTO_COMP	R/W	1h	This setting is used to allocate the IO pad for TX_CLK for doing the Auto compensation for the samlping of TXEN & TXDATA. 0 : Manual Compensation using CLK_IN no TX_CLK Pad, IP input is tied to '0'. 1: Auto Compensation based on sampling of TX_CLK. Pad is allocated. Reset type: ECAT.XRSn
5-4	RESERVED	R/W	0h	Reserved
3-2	PHY_PORT_CNT	R/W	1h	Indicates the number of PHY ports selected for operation inaddition to Port0 which is default (information only, doesn't change configuration) 00-One port operation (Port0) 01-Two port operation(Port0,Port1) 10-Three port operation(Port0,Port1,Port2) : Reserved 11-Four port operation(Port0,Port1,Port2,Port3): Reserved Programming reserved configuration causes selection of Reset value. Reset type: ECAT.XRSn
1-0	RESERVED	R/W	0h	Reserved

35.6.3.4 ESCSS_SYNC_IO_CONFIG Register (Offset = Ch) [Reset = 00000088h]

ESCSS_SYNC_IO_CONFIG is shown in Figure 35-42 and described in Table 35-44.

Return to the Summary Table.

SYNC0/1 IO configurations including enable & Pad Select

Figure 35-42. ESCSS_SYNC_IO_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
WRITE_KEY							
R-0/W-0h							
7	6	5	4	3	2	1	0
SYNC1_GPIO_EN	RESERVED	RESERVED		SYNC0_GPIO_EN	RESERVED	RESERVED	
R/W-1h	R-0-0h	R/W-0h		R/W-1h	R-0-0h	R/W-0h	

Table 35-44. ESCSS_SYNC_IO_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-8	WRITE_KEY	R-0/W	0h	The key value should be 0xa5 for the writes to bit 0 take effect. Writes of other values will be ignored. Reset type: ECAT.XRSn
7	SYNC1_GPIO_EN	R/W	1h	Enables the direct mux between Sync1 output of EtherCAT & other GPIO functions. Reset type: ECAT.XRSn
6	RESERVED	R-0	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3	SYNC0_GPIO_EN	R/W	1h	Enables the direct mux between Sync0 output of EtherCAT & other GPIO functions. Reset type: ECAT.XRSn
2	RESERVED	R-0	0h	Reserved
1-0	RESERVED	R/W	0h	Reserved

35.6.3.5 ESCSS_LATCH_IO_CONFIG Register (Offset = 10h) [Reset = 00000088h]

ESCSS_LATCH_IO_CONFIG is shown in Figure 35-43 and described in Table 35-45.

Return to the Summary Table.

LATCH0/1 IO configurations including enable & Pad Select

Figure 35-43. ESCSS_LATCH_IO_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
WRITE_KEY							
R-0/W-0h							
7	6	5	4	3	2	1	0
LATCH1_GPIO_EN	RESERVED	RESERVED		LATCH0_GPIO_EN	RESERVED	RESERVED	
R/W-1h	R-0-0h	R/W-0h		R/W-1h	R-0-0h	R/W-0h	

Table 35-45. ESCSS_LATCH_IO_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-8	WRITE_KEY	R-0/W	0h	The key value should be 0xa5 for the writes to bit 0 take effect. Writes of other values will be ignored. Reset type: ECAT.XRSn
7	LATCH1_GPIO_EN	R/W	1h	Enables the direct mux between LATCH1 input from IOPAD & other GPIO functions to the EtherCATSS input Reset type: ECAT.XRSn
6	RESERVED	R-0	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3	LATCH0_GPIO_EN	R/W	1h	Enables the direct mux between LATCH0 input from IOPAD & other GPIO functions to the EtherCATSS input Reset type: ECAT.XRSn
2	RESERVED	R-0	0h	Reserved
1-0	RESERVED	R/W	0h	Reserved

35.6.3.6 ESCSS_GPIN_SEL Register (Offset = 14h) [Reset = 00000000h]

ESCSS_GPIN_SEL is shown in Figure 35-44 and described in Table 35-46.

Return to the Summary Table.

Register to configure each GPI input is connected to IO-pad or not.

Figure 35-44. ESCSS_GPIN_SEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIN_SEL																															
R/W-0h																															

Table 35-46. ESCSS_GPIN_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPIN_SEL	R/W	0h	Allows bit-wise selection of the GPIN be connected from GPIO PAD. Once those are not driven by GPIO, will be driven from register writable from local Host. 0: No connection to GPIO PAD, but connects to ESCSS_GPIN_DAT. 1: Mux Select the GPIN from the dedicated IO PAD. This acts as Mux select for input from GPIO over tieoff. Reset type: ECAT.XRSn

35.6.3.7 ESCSS_GPOUT_SEL Register (Offset = 1Ch) [Reset = 00000000h]

ESCSS_GPOUT_SEL is shown in Figure 35-45 and described in Table 35-47.

Return to the Summary Table.

Register to configure each GPO to be connected to IO-pad or not.

Figure 35-45. ESCSS_GPOUT_SEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPOUT_SEL																															
R/W-0h																															

Table 35-47. ESCSS_GPOUT_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPOUT_SEL	R/W	0h	Allows bit-wise selection for GPOUT connection to IO PAD. hence acts as direct mux select between GPO & other non-EtherCAT functions. 0: GPO is not connected to dedicated IO instead non-EtherCAT function is connected. 1: Connect the GPOUT to the dedicated IO pad through output buffer. Reset type: ECAT.XRSn

35.6.3.8 ESCSS_LED_CONFIG Register (Offset = 24h) [Reset = 00000000h]

ESCSS_LED_CONFIG is shown in Figure 35-46 and described in Table 35-48.

Return to the Summary Table.

Register to select of LED o/p is connected to IO-PAD

Figure 35-46. ESCSS_LED_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED		RESERVED	RUN	ERR	STATE	RESERVED	RESERVED
R/W-0h		R-0-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 35-48. ESCSS_LED_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R/W	0h	Reserved
5	RESERVED	R-0	0h	Reserved
4	RUN	R/W	0h	Acts as Mux select to enable RUN LED function directly onto output pad. 0: The non-EtherCAT function is selected on the IO 1: RUN LED is selected to be output on the IO This selection assumes both buffer input and buffer enable connection as required. Reset type: ECAT.XRSn
3	ERR	R/W	0h	Acts as Mux select to enable ERR LED function directly onto output pad. 0: The non-EtherCAT function is selected on the IO 1: ERR LED is selected to be output on the IO This selection assumes both buffer input and buffer enable connection as required. Reset type: ECAT.XRSn
2	STATE	R/W	0h	Acts as Mux select to enable STATE LED function directly onto output pad. 0: The non-EtherCAT function is selected on the IO 1: STATE LED is selected to be output on the IO This selection assumes both buffer input and buffer enable connection as required. Reset type: ECAT.XRSn
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

35.6.3.9 ESCSS_MISC_CONFIG Register (Offset = 28h) [Reset = 00000000h]

ESCSS_MISC_CONFIG is shown in Figure 35-47 and described in Table 35-49.

Return to the Summary Table.

Configuration info for the MII interface containing TX_SHIFT compensation values, PHY Address offset, EEPROM SIZE etc.

Figure 35-47. ESCSS_MISC_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED					PHY_ADDR		
R-0-0h					R/W-0h		
7	6	5	4	3	2	1	0
PHY_ADDR		PDI_EMULATION	EEPROM_SIZE	TX1_SHIFT_CONFIG		TX0_SHIFT_CONFIG	
R/W-0h		R/W-0h	R/W-0h	R/W-0h		R/W-0h	

Table 35-49. ESCSS_MISC_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	Reserved
10-6	PHY_ADDR	R/W	0h	These bits will be hooked up to the PHY_OFFSET[4:0] input of the EtherCAT IP. Reset type: ECAT.XRSn
5	PDI_EMULATION	R/W	0h	This bit will be hooked up to the PDI_EMULATION input of the EtherCAT IP. Reset type: ECAT.XRSn
4	EEPROM_SIZE	R/W	0h	This bit will be hooked up to the EEPROM_SIZE input of the EtherCAT IP . This is set to 0 for EEPROMs of size 16K bits or lower. This is set to 1 for EEPROMs of size above 16K bits. Reset type: ECAT.XRSn
3-2	TX1_SHIFT_CONFIG	R/W	0h	Two bit TX_SHIFT configuration in terms of 10ns counts for port0. This is the shift added to TX_ENA & TX_DATA to match delay of PHY TX_CLK w.r.t. device internal clock. Reset type: ECAT.XRSn
1-0	TX0_SHIFT_CONFIG	R/W	0h	Two bit TX_SHIFT configuration in terms of 10ns counts for port0. This is the shift added to TX_ENA & TX_DATA to match delay of PHY TX_CLK w.r.t. device internal clock. Reset type: ECAT.XRSn

Chapter 36

Fast Serial Interface (FSI)

This chapter contains a general description of the Fast Serial Interface (FSI) module. The FSI is a serial peripheral capable of reliable high-speed communication across isolation barriers.

36.1 Introduction

The Fast Serial Interface (FSI) module is a serial communication peripheral capable of reliable high-speed communication across isolation devices. Galvanic isolation devices are used in situations where two different electronic circuits, which do not have common power and ground connections, must exchange information. Though isolation devices facilitate these signal communications, isolation devices can also introduce a large delay on the signal lines and add skew between the signals. The FSI is designed specifically to make sure reliable high-speed communication for system scenarios that involve communication across isolation barriers without adding components.

The FSI consists of independent transmitter (FSITX) and receiver (FSIRX) cores. The FSITX and FSIRX cores are configured and operated independently.

For additional information on the FSI module, refer to *Fast Serial Interface (FSI) Skew Compensation*.

36.1.1 FSI Related Collateral

Foundational Materials

- C28x Academy - FSI
- C29x Academy - Fast Serial Interface (FSI)

Getting Started Materials

- *Fast Serial Interface (FSI) Skew Compensation Application Report*
- Fast serial interface (FSI) adapter board evaluation module
- *Using the Fast Serial Interface (FSI) With Multiple Devices in an Application Application Report*

Expert Materials

- *Design Guide: TIDM-02006 Distributed Multi-axis Servo Drive Over Fast Serial Interface (FSI) Reference Design*
- *The Essential Guide for Developing With C2000 Real-Time Microcontrollers Application Report*
  - Refer to the See sections 'Distributed Real-Time Control Across an Isolation Boundary' and 'Solving Event Synchronization Across Multiple Controllers in Decentralized Control Systems'. section

36.1.2 FSI Features

The FSI module includes the following features:

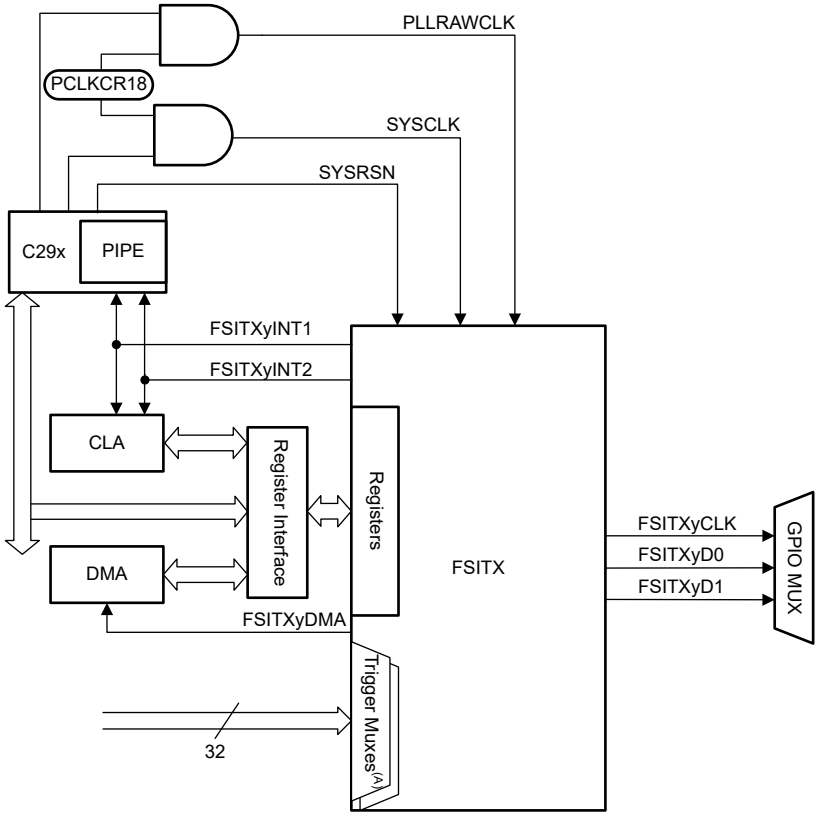
- Independent transmitter and receiver cores
- Source-synchronous transmission
- Double Data Rate (DDR)
- One or two data lines
- Programmable data length
- Skew adjustment block to compensate for board and system delay mismatches
- Frame error detection
- Programmable frame tagging for message filtering
- Hardware ping to detect line breaks during communication (ping watchdog)
- Two interrupts per FSI core
- Externally-triggered frame generation
- Hardware- or software-calculated CRC
- Embedded ECC computation module
- Register write protection
- FSI-SPI compatibility mode (limited features available)
- Tag match notifications

36.2 System-level Integration

This section describes the device-level integration of the FSI module. Some of the features can require additional configuration of modules that are not within the scope of this chapter, the details can be found elsewhere in this TRM.

36.2.1 CPU Interface

The following diagrams show the CPU interface of each FSI module.



A. The signals connected to the trigger muxes are described in Section 36.2.5.

Figure 36-1. FSI Transmitter (FSITX) CPU Interface

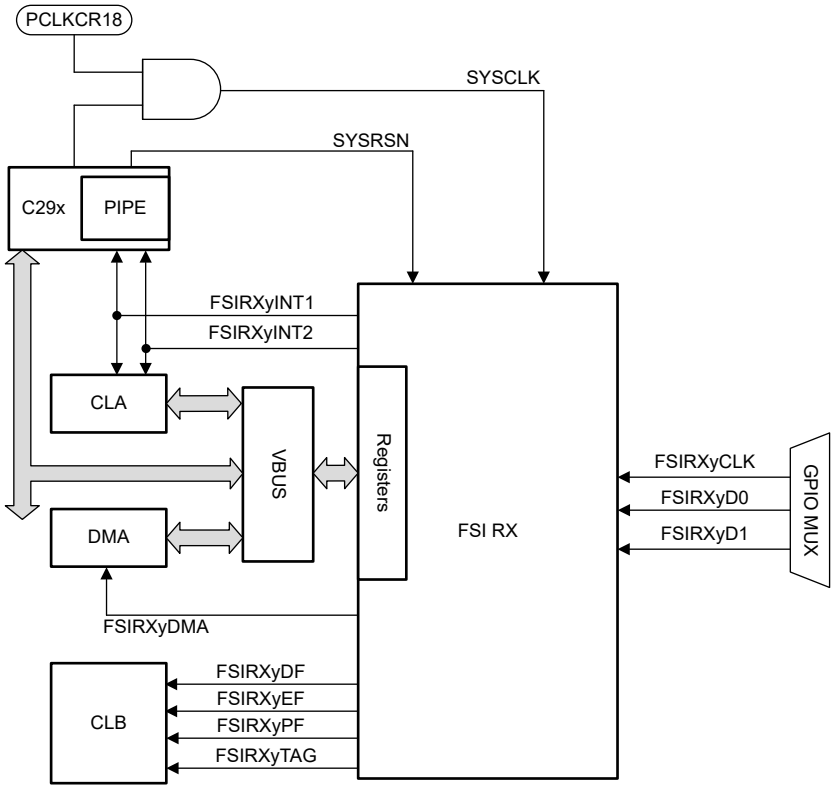


Figure 36-2. FSI Receiver (FSIRX) CPU Interface with CLB

36.2.2 Signal Description

FSI is a point-to-point communication protocol. Hence, an FSI transmitter core communicates directly to a single FSI receiver core. Similarly, an FSI receiver core receives data from a single FSI transmitter core.

Each FSI core has three signals: one clock and two data signals. Data is always transmitted or received with the most-significant bit of each frame field being first. If multilane transmissions are not used, the TXD1 and RXD1 signals can be left unconnected and the GPIOs repurposed for other application needs. Table 36-1 and Table 36-2 describe the various signals that can be selected by the PADCONFIG register to be brought out to device pins.

CAUTION
The maximum RXCLK rate is SYSCLK/2 and must not exceed this limit.

Table 36-1. FSI Receiver Core Signals

Signal Name	Direction	Description	Inactive Level(1)
RXCLK	Input	This is the receive clock input signal for the FSI receive module. This must be connected to TXCLK of the transmitting FSI module.	Logic High
RXD0	Input	This is the primary data input line for reception. This must be connected to the TXD0 of the transmitting FSI module.	Logic High
RXD1	Input	This is an additional data input line for reception. This signal must be connected to the TXD1 of the transmitting FSI module to use multilane transmission.	Logic High

(1) Inactive level refers to the state of the pin while the module is not actively receiving data.

Table 36-2. FSI Transmitter Core Signals

Signal Name	Direction	Description	Inactive Level(1)
TXCLK	Output	This is the transmit clock and is driven by the FSI transmit module. During a transmission, four clock edges are transmitted before the start of frame phase (preamble) and four clock edges follow the last bit of the frame (postamble). Data is transmitted on both edges of the clock. In FSI-SPI compatibility mode, the preamble and the post frame clock edges are not transmitted. Data is transmitted only on one edge of the clock. Data transmits on rising edge and received on falling edge of the clock.	Logic High
TXD0	Output	This is the primary data output line for transmission and is driven by the FSI transmit module. When the FSI is configured for multilane transmission, TXD0 contains all the even numbered bits of the data and CRC bytes. Other frame fields such as frame type, start-of-frame, tag, and end-of-frame are transmitted in full.	Logic High
TXD1	Output	This is an additional data output line for transmission, if the FSI is configured for multilane transmission. This signal is driven by the FSI transmit module. During transmission, the data bits are split between TXD0 and TXD1. TXD1 contains all the odd numbered bits of the data and CRC bytes. This applies only to the data words and the CRC bytes. Other data frame related information like Frame Type, Start-of-Frame, Tag and End-of-frame, the state of this line are identical to TXD0.	Logic High

(1) Inactive level refers to the state of the pin while the module is not actively transmitting, or held in reset.

36.2.2.1 Configuring Device Pins

The GPIO mux registers must be configured to connect this peripheral to the device pins. To avoid glitches on the pins, the GPyGMUX bits must be configured first (while keeping the corresponding GPyMUX bits at the default of zero), followed by writing the GPyMUX register to the desired value.

Some IO functionality is defined by GPIO register settings independent of this peripheral. For input signals, the GPIO input qualification must be set to asynchronous mode by setting the appropriate GPxQSELn register bits to 0x3. The internal pullups can be configured in the GPyPUD register. See the *General Purpose Input-Output (GPIO)* chapter for more details on the GPIO mux and settings.

36.2.3 FSI Interrupts

Each FSI module contains multiple interrupt sources that can be assigned to two different interrupt vectors: INT1 and INT2. Each interrupt source has an associated status flag, force, and clear bits in the EVT_STS, EVT_FRC, and the EVT_CLR registers, respectively.

Each interrupt can be assigned to either interrupt vector, INT1 and INT2, to allow for two priority levels. Alternately, the interrupt source can be prevented from generating any interrupt, though the status flag can still be set and monitored by software. The transmitter events are assigned to either interrupt vector in the TX_INT_CTRL register. The receiver events are assigned an interrupt vector using RX_INT1_CTRL and RX_INT2_CTRL registers. If an interrupt is not required, make sure the bit is not set in the respective INT_CTRL register.

36.2.3.1 Transmitter Interrupts

The transmitter can generate the following interrupts:

- **Frame Done (FRAME_DONE):** This event indicates that FSI has completed transmitting a frame.
- **Buffer Underrun (BUF_UNDERRUN):** This event indicates that the transmit buffer has experienced underrun. Buffer underrun occurs when the transmitter tries to read data from a location which has not yet be written to by the CPU, or RTDMA.
- **Buffer Overrun (BUF_OVERRUN):** The buffer overrun interrupt is generated when the buffer has experienced overrun. Buffer overrun can occur if a piece of data is overwritten before the data has been transmitted.
- **Ping Frame Triggered (PING_TRIGGERED):** The ping frame triggered interrupt is generated when the ping frame has been triggered. This bit is set when the ping counter has timed out or an external ping trigger event has occurred.

36.2.3.2 Receiver Interrupts

The receiver core is capable of generating interrupts from many different events:

- **Ping Watchdog Timeout (PING_WD_TO):** This event indicates that the ping watchdog timer has timed out. The receiver has not received a valid frame within the time period specified in the RX_PING_WD_REF register.
- **Frame Watchdog Timeout (FRAME_WD_TO):** This event indicates that the frame watchdog timer has timed out. The conditions of this timeout are set using the RX_FRAME_WD_CTRL register. As soon as the start of frame phase is detected, the frame watchdog counter starts counting from 0. The end of frame phase must complete by the time the watchdog counter reaches the reference value. If this does not happen, the watchdog times out and this event is generated. If this event occurs, the receiver must undergo a soft reset and subsequent resynchronization to resume proper operation.
- **CRC Error (CRC_ERR):** This error indicates that a CRC error has occurred. A CRC error is generated when the received CRC and the computed CRC do not match.
- **Frame Type Error (TYPE_ERR):** This error indicates that an invalid frame type has been received. If this error occurs, the receiver must undergo a soft reset and subsequent resynchronization to resume proper operation.
- **End-of-Frame Error (EOF_ERR):** This error indicates that an invalid end-of-frame bit pattern has been received. If this error occurs, the receiver must undergo a soft reset and subsequent resynchronization to resume proper operation.
- **Receive Buffer Overrun (BUF_OVERRUN):** This event indicates that an overrun condition has occurred in the receive buffer.
- **Receive Buffer Underrun (BUF_UNDERRUN):** This event indicates that an underrun condition has occurred in the receive buffer. This condition occurs when software reads an empty buffer.
- **Frame Done (FRAME_DONE):** This event indicates that a valid frame has been received without error.
- **Error Frame Received (ERR_FRAME):** This event indicates that an error frame has been received.
- **Ping Frame Received (PING_FRAME):** This event indicates that a ping frame has been received.
- **Frame Overrun (FRAME_OVERRUN):** This event indicates that a new frame has been received while the FRAME_DONE flag was still set.
- **Data Frame Received (DATA_FRAME):** This event indicates that a data frame has been received.
- **Ping Tag Matched (PING_TAG_MATCH):** This event indicates that a ping frame with a matching tag has been received.
- **Data Tag Matched (DATA_TAG_MATCH):** This event indicates that a data frame with a matching tag has been received.
- **Error Tag Matched (ERROR_TAG_MATCH):** This event indicates that an error frame with a matching tag has been received.

36.2.3.3 Configuring Interrupts

To configure interrupts on the FSI, the application must select the interrupt vector for each desired event using the TX_INT_CTRL register for the transmitter, and RX_INT1_CTRL and RX_INT2_CTRL registers for the receiver. There is no module-level interrupt enable bit to configure.

Note

If an event is registered for both interrupt vectors, both interrupts fire. There are no hardware checks for overlapping interrupt vector assignments.

36.2.3.4 Handling Interrupts

Inside the interrupt service routine (ISR), the user must clear the event flag using the EVT_CLR register and then acknowledge the CPU interrupt.

If the one event occurs multiple times before the corresponding bit is cleared by software, no new interrupt is generated.

If multiple events occur simultaneously, or very close in time, it is possible to handle multiple conditions within a single interrupt. Each flag is independently set by hardware and must be cleared by application software. If multiple different events occur, the ISR can handle each in whatever order is deemed necessary by the application. It is not advisable to clear the full interrupt status register in every ISR. This can cause the application to miss events that can be detrimental to the application. A sample sequence for handling interrupts on the receiver follows; the transmitter routine is similar.

- On receiving an interrupt, copy the current state of the receive event and error status flag register (RX_EVT_STS) into a local snapshot variable.
- Read all of the bits from the snapshot to determine the events that require action.
- Perform the necessary actions for each of the events seen in the snapshot.
- Write to the receive event and error clear register (RX_EVT_CLR) with the snapshot to clear only those interrupts that were set at the beginning of the ISR.
- Repeat this sequence for every generated ISR.

There is a chance that another event occurred during the just-handled ISR since only the snapshot of events was handled and then cleared; an event flag can still be set at the end of the ISR. As soon as the ISR completes, a new interrupt is generated and this flag is still set and can be handled accordingly.

Software accesses tied to multiple events and handled within the same ISR can cause race conditions that cause the software to not function as desired. For example, it is recommended to use different interrupt lines if the user wants to enable events for both ping and data frames. If both events are handled within the same interrupt line, the software can only respond to one of the events if both events occur close in time.

36.2.4 RTDMA Interface

Both the transmitter and receiver are capable of using the RTDMA for automatic data transfers. The RTDMA trigger is independent from the interrupt signals. RTDMA events are only triggered on the completion of a data frame.

The transmitter RTDMA trigger is enabled by setting TX_DMA_CTRL.DMA_EVT_EN to 1. The transmitter must also set TX_OPER_CTRL_LO.START_MODE to 0x2 to allow either a write to the TX_FRAME_CTRL.START bit or to the TX_FRAME_TAG_UDATA register to start the transmission.

The receiver RTDMA trigger is enabled by setting RX_DMA_CTRL.DMA_EVT_EN to 1.

Refer to Section 36.3.2 and Section 36.3.3 for more RTDMA information specific to each FSI Module.

36.2.5 External Frame Trigger Mux

The FSI has two muxes connected to the transmitter module. These muxes are used to select triggers to start ping frames, and generic frames. These muxes are independently configured for each type of frame. The application can select one trigger source per frame type. Use of these triggers are optional.

The external ping frame trigger is configured by setting TX_PING_CTRL.EXT_TRIG_SEL to the index of the desired trigger. TX_PING_CTRL.EXT_TRIG_EN must also be set to allow the trigger to generate a ping frame.

The generic frame trigger is configured by setting TX_OPER_CTRL_HI.EXT_TRIG_SEL to the index of the desired trigger. TX_OPER_CTRL_LO.START_MODE must be set to 0x1 for a frame to be transmitted by an external trigger.

Table 36-3. External Trigger Sources and Their Index

Index	External Trigger Source
0	EPWMXBAR1
1	EPWMXBAR2
2	EPWMXBAR3
3	EPWMXBAR4
4	EPWMXBAR5
5	EPWMXBAR6
6	EPWMXBAR7
7	EPWMXBAR8
8	EPWM1_SOCA
9	EPWM1_SOCB
10	EPWM2_SOCA
11	EPWM2_SOCB
12	EPWM3_SOCA
13	EPWM3_SOCB
14	EPWM4_SOCA
15	EPWM4_SOCB
16	EPWM5_SOCA
17	EPWM5_SOCB
18	EPWM6_SOCA
19	EPWM6_SOCB
20	EPWM7_SOCA
21	EPWM7_SOCB
22	EPWM8_SOCA
23	EPWM8_SOCB
24	EPWM9_SOCA
25	EPWM9_SOCB
26	EPWM10_SOCA
27	EPWM10_SOCB
28	EPWM11_SOCA
29	EPWM11_SOCB
30	EPWM12_SOCA
31	EPWM12_SOCB
32	EPWM13_SOCA
33	EPWM13_SOCB
34	EPWM14_SOCA

Table 36-3. External Trigger Sources and Their Index (continued)

Index	External Trigger Source
35	EPWM14_SOCB
36	EPWM15_SOCA
37	EPWM15_SOCB
38	EPWM16_SOCA
39	EPWM16_SOCB
40	CLB1_OUT30
41	CLB1_OUT31
42	CLB2_OUT30
43	CLB2_OUT31
44	CLB3_OUT30
45	CLB3_OUT31
46	CLB4_OUT30
47	CLB4_OUT31
48	CLB5_OUT30
49	CLB5_OUT31
50	CLB6_OUT30
51	CLB6_OUT31
52	ADCSOCA
53	ADCSOCB
54	CPU1_TINT0
55	CPU1_TINT1
56	CPU1_TINT2
57	CPU2_TINT0
58	CPU2_TINT1
59	CPU2_TINT2
60	CPU3_TINT0
61	CPU3_TINT1
62	CPU3_TINT2
63	RTDMA1_CH1
64	RTDMA1_CH2
65	RTDMA1_CH3
66	RTDMA1_CH4
67	RTDMA1_CH5
68	RTDMA1_CH6
69	RTDMA1_CH7
70	RTDMA1_CH8
71	RTDMA1_CH9
72	RTDMA1_CH10
73	RTDMA2_CH1
74	RTDMA2_CH2
75	RTDMA2_CH3
76	RTDMA2_CH4
77	RTDMA2_CH5
78	RTDMA2_CH6
79	RTDMA2_CH7

Table 36-3. External Trigger Sources and Their Index (continued)

Index	External Trigger Source
80	RTDMA2_CH8
81	RTDMA2_CH9
82	RTDMA2_CH10
83-86	Reserved
87	FSIA/B/C/D RX_TRIG0 (TXA-->RXA, TXB-->RXB, TXC-->RXC, TXD-->RXD)
88	FSIA/B/C/D RX_TRIG1 (TXA-->RXA, TXB-->RXB, TXC-->RXC, TXD-->RXD)
89	FSIA/B/C/D RX_TRIG2 (TXA-->RXA, TXB-->RXB, TXC-->RXC, TXD-->RXD)
90	FSIA/B/C/D RX_TRIG3 (TXA-->RXA, TXB-->RXB, TXC-->RXC, TXD-->RXD)
91-94	Reserved
95	EPWM17_SOCA
96	EPWM17_SOCB
97	EPWM18_SOCA
98	EPWM18_SOCB
99-127	Reserved

36.3 FSI Functional Description

36.3.1 Introduction to Operation

The Fast Serial Interface Transmitter and Receiver modules (FSI_TX/FSI_RX) are two completely independent modules on the device. Each module has an independent set of control registers, clocking, and interrupts. The following sections describe the frame format and the various initialization and configuration procedures for both the transmitter and receiver.

36.3.2 FSI Transmitter Module

The FSI transmitter module handles the framing of data, CRC generation, and signal generation of TXCLK, TXD0, and TXD1, as well as interrupt generation. The operation of the transmitter core is controlled and configured through programmable control registers. The transmitter control registers allow the CPU to program, control, and monitor the operation of the FSI receiver. The transmit data buffer is accessible by the CPU and the RTDMA.

The transmitter has the following features:

- Automated ping frame generation
- Externally triggered ping frames
- Externally triggered data frames
- Software-configurable frame lengths
- 16-word data buffer
- Data buffer underrun and overrun detection
- Hardware-generated CRC on data bits
- Software ECC calculation on select data
- RTDMA support

Figure 36-3 shows the high-level block diagram of the FSI transmitter. Figure 36-4 shows the block diagram of the transmitter core submodule.

The following sections describe the various aspects of the FSI transmitter in detail.

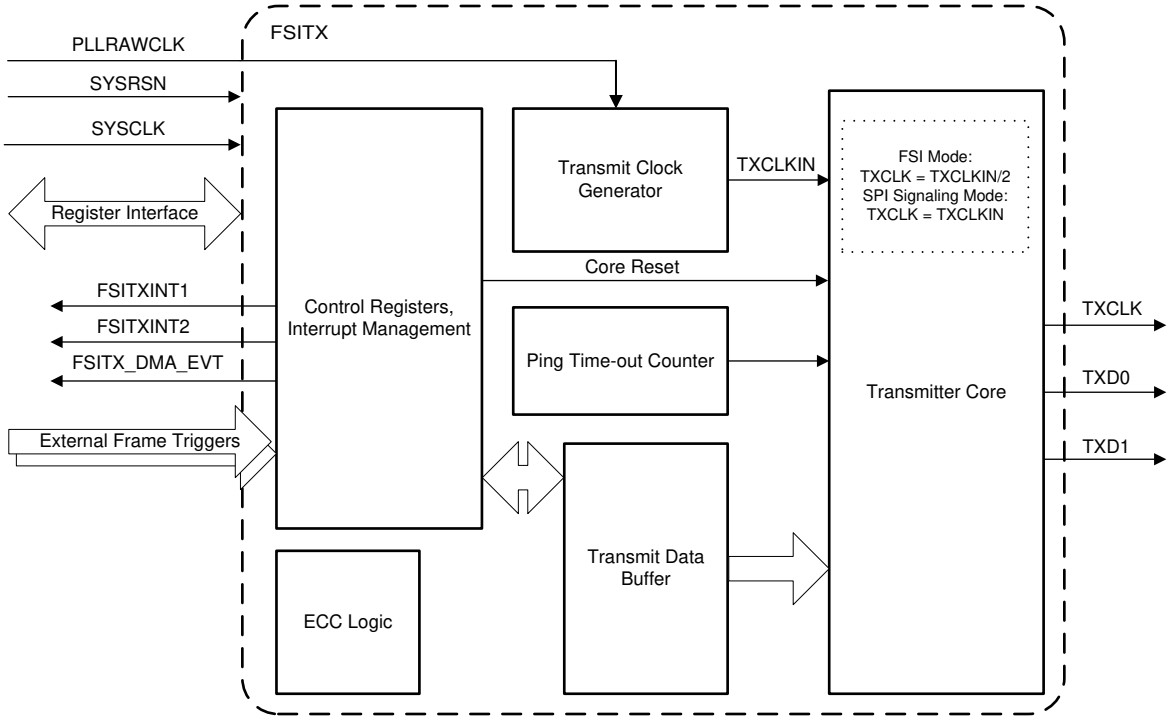


Figure 36-3. FSI Transmitter Block Diagram

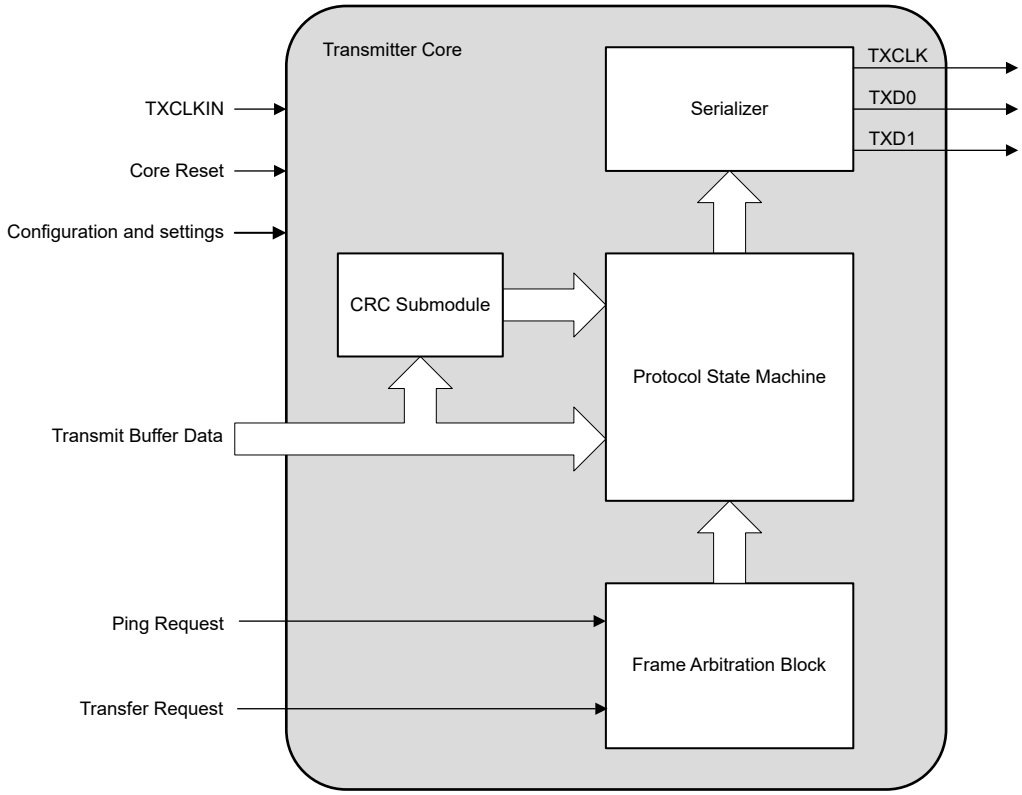


Figure 36-4. FSI Transmitter Core Block Diagram

36.3.2.1 Initialization

On the first initialization or after a module reset due to an underrun condition, the transmitter module executes the following initialization sequence to start or resume transmit operations.

1. Initialize the transmitter clock by setting TX_CLK_CTRL.CLK_RST to 1 and subsequently clearing the bit.
2. Set the clock to the transmitter core to PLLRAWCLK by setting TX_OPER_CTRL_LO.SEL_PLLCLK to 1.
3. Set the clock prescaler value to the desired rate by writing to TX_CLK_CTRL.PRESCALE_VAL.
4. Enable the transmitter clock divider by setting TX_CLK_CTRL.CLK_EN to 1.
5. Assert the transmitter module soft reset by writing 0xA501 to TX_MAIN_CTRL.
6. Wait four TXCLK cycles.
7. Release the transmitter core from reset by writing 0xA500 to TX_MAIN_CTRL.

After initialization and configuration, the transmitter module synchronizes with the receiver module before transmitting. The synchronization sequence is described in Section 36.4.1.

CAUTION

Do not change TX_CLK_CTRL.PRESCALE_VAL while the clock is enabled (TX_CLK_CTRL.CLK_EN = 1). Doing so can cause undefined behavior.

36.3.2.2 FSI_TX Clocking

The transmitter core registers and control logic run off of the device system clock (SYSCLK).

The FSI Transmit Clock (TXCLK) is derived from PLLRAWCLK. PLLRAWCLK is divided down by configuring the clock prescaler value (TX_CLK_CTRL.PRESCALE_VAL) then setting the clock divider enable bit (TX_CLK_CTRL.CLK_EN). The clock prescaler value can be set to divide PLLRAWCLK by 1 (TX_CLK_CTRL.PRESCALE_VAL = 0x0 or 0x1) through 255(TX_CLK_CTRL.PRESCAL_VAL = 0xFF). Though TXCLK and SYSCLK are both derived from PLLRAWCLK, TXCLK is asynchronous with respect to SYSCLK.

CAUTION

TXCLK must never be configured to be faster than SYSCLK/2.

36.3.2.3 Transmitting Frames

On the transmitter, the ping frame is the only frame that can be set up and transmitted without any further software or RTDMA intervention. Ping frames can be transmitted by any (or all) of the three sources: automatic ping timer, software, or external triggers.

Each available frame type can be sent multiple ways. Generically, the following steps must be executed before the frame is sent. These steps can be executed in any order before the start condition is set.

1. Configure the frame type
2. Set the frame tag
3. If the frame to be sent is a data frame:
   - Set the user data
   - Write to the data buffer
   - Set the word length if the frame is a software defined frame length
4. Set the start condition

Note

Transmit Frame Start Restriction:

A new frame transmission can be initiated by one of the methods selected in the TX_OPER_CTRL_LO.START_MODE bits. If there is already a PING frame transmission taking place, due to a hardware initiated PING timer, the new frame transmission begins as soon as the on-going PING transmission is completed.

Once a START of frame has been initiated, the next START of frame is recognized when the first frame has started transmitting the End-of-Frame (EOF) field. If a new START trigger arrives before the current transmission has reached the EOF field, the trigger is lost without a notification.

Note

There is no hardware check implemented to check whether the type field written by software is valid or not. If an invalid type is used and a frame transmission is initiated, the behavior is:

- The transmitted frame structure is exactly like an NWORD data frame. The size of the data frame is determined by the value in the TX_FRAME_CTRL.N_WORDS register.
- The frame type field of the transmitted data frame is transmitted as programmed. If this is received by an FSI receiver, a Type error is generated.

This mechanism can be used for force a Type error in a received frame for testing purposes.

The following sections describe the specific configuration for each frame type and start condition.

36.3.2.3.1 Software Triggered Frames

The most basic way to transmit a data frame is through software. Each step must be handled by the application. To send a data frame using software, the following steps must be executed. Steps 1-6 can be executed in any order before setting TX_FRAME_CTRL.START. Some fields do not need to be reconfigured for every transmission. The frame tag, user data, and frame type are sticky and are retransmitted in the subsequent frame unless modified by software.

1. Write the data to be transmitted to the next location of the transmit data buffer.
2. Set TX_FRAME_CTRL.FRAME_TYPE to the appropriate value for the type of frame to be transmitted.
3. Set TX_FRAME_CTRL.N_WORDS to 1 less than the number of words to be transmitted if TX_FRAME_CTRL.FRAME_TYPE is set to 0011, the frame type of the software-defined length data frame. That is, if 16 words are transmitted, N = 16, set TX_FRAME_CTRL.N_WORDS to 15.
4. When the frame is assembled before transmitting, the FSITX hardware calculates the CRC to be transmitted. If TX_OPER_CTRL_LO.SW_CRC is 1, the application can calculate a custom CRC value and then set TX_USER_CRC to the result.
5. Set TX_FRAME_TAG_UDATA.FRAME_TAG to the desired tag.
6. Set TX_FRAME_TAG_UDATA.USER_DATA to the desired user data.
7. Set TX_FRAME_CTRL.START to 1 to initiate the transmission of the data frame.

Once the frame transmission has started, the TX_FRAME_CTRL.START is cleared by hardware. To monitor if the frame has completed, the software can poll TX_EVT_STS.FRAME_DONE.

36.3.2.3.2 Externally Triggered Frames

The transmitter can transmit frames when triggered by an external source. See Section 36.2.5 for more information on the available external triggers.

To transmit frames using an external trigger, the application must follow the same procedure as described in Section 36.3.2.3.1. The only difference is that in Step 7, the start condition is automatically set when the external trigger condition is met rather than by software.

Note that by externally triggering frames, the frame information to be sent is pulled from the same registers described in the previous section. Because of this, it is possible to send any type of frame from an external trigger including ping, error, and data frames. Also, there is no hardware mechanism by which the FSI can determine if multiple triggers occur. The FSITX takes the data as is, and the application software makes sure that this data has been updated as necessary.

Using TX_EVT_STS fields either by polling or by interrupts, the application can populate or update the frame information to be sent in the next frame

36.3.2.3.3 Ping Frame Generation

Assuming the FSI transmitter has already been properly initialized, the following sequences can be used to configure and send ping frames.

36.3.2.3.3.1 Automatic Ping Frames

To generate periodic ping frames, the following steps must be followed:

1. Initialize the ping counter by writing 1 to TX_PING_CTRL.CNT_RST.
2. Set the desired ping tag to TX_PING_TAG.TAG.
3. Set the ping timer reference value to TX_PING_TO_REF.TO_REF.
4. Enable the ping timer by writing 1 to TX_PING_CTRL.TIMER_EN.

The ping timer is a free-running counter that counts up from 0. The current value of the ping timer counter is found in TX_PING_TO_CNT. When the current value of TX_PING_TO_CNT matches the reference value TX_PING_TO_REF.TO_REF, the TX_EVT_STS.PING_TRIGGERED is set. TX_PING_TO_CNT resets to 0 and resumes counting until the next match has occurred or the ping timer is halted by software (TX_PING_CTRL.TIMER_EN is set to 0).

36.3.2.3.3.2 Software Triggered Ping Frame

Software can also manually generate a ping frame. The process for sending a ping frame with software is very similar to sending the other types of frames. The following steps must be followed:

1. Set TX_FRAME_CTRL.FRAME_TYPE to 0000’b to denote that the frame being sent is a Ping Frame.
2. Set TX_FRAME_TAG_UDATA.FRAME_TAG to the desired value.
3. Write 1 to TX_FRAME_CTRL.START. This starts the transmission.

Once the frame transmission has started, the TX_FRAME_CTRL.START is cleared by hardware. To monitor if the frame has completed, the software can poll TX_EVT_STS.FRAME_DONE.

36.3.2.3.3.3 Externally Triggered Ping Frame

The last source for generating ping frames is an external trigger. One of up to 32 different triggers can be selected. See Section 36.2.5 for the list of input sources.

CAUTION

Ping frames can be triggered by both an external trigger source and the internal ping timer. If TX_PING_CTRL.EXT_TRIG_EN is set to 1, the external trigger source takes precedence and the ping timer is ignored.

36.3.2.3.4 Transmitting Frames with RTDMA

The FSI transmitter can send data that is continuously applied with the RTDMA. A RTDMA trigger is generated every time a data frame transmission is completed. This is concurrent with the FRAME_DONE signal that sets the TX_EVT_STS.FRAME_DONE flag.

To transmit continuous data with the RTDMA, some configurations need to be made on the transmitter:

First, set TX_DMA_CTRL.DMA_EVT_EN to 1. This allows the RTDMA trigger to propagate to the RTDMA module. Next, TX_OPER_CTRL_LO.START_MODE must be set to 0x2. The transmitter is now able to start a transmission using a software write to TX_FRAME_CTRL.START or TX_FRAME_TAG_UDATA..

The RTDMA must also be configured properly for the FSI to send the data. One way of using the RTDMA to continuously feed the transmit buffer is:

- Set up two RTDMA channels to be triggered by the same FSI transmitter and RTDMA trigger.
- Configure one channel to fill the transmit buffer.
- Configure the other channel to set the frame tag and user data fields
- Since the FSI transmit buffer is a 16-word circular buffer, make sure the RTDMA channel servicing the data buffer wraps the after 16 words are copied.

Note

Because the frame tag and user data must be written in to initiate the transmission of the frame, use two consecutive RTDMA channels. This makes sure that the RTDMA channels are always executed in sequence. The RTDMA channel servicing the data buffer must be the lower numbered channel and the tag/user data channel must be the next. For example, configure RTDMA channel 3 to service the data buffer, and configure RTDMA channel 4 to service the tag and user data.

36.3.2.4 Transmit Buffer Management

The FSI transmitter has a 16-word buffer that the FSI transmitter pulls data to transmit. This buffer is implemented as a circular buffer, not a FIFO, so some care must be taken to properly interpret buffer overrun and underrun, as well as the TX_BUF_PTR_STS register. These flags and pointers work under the assumption that the software or RTDMA is using the buffer as a circular buffer. This mode of operation is the only way that the overrun, underrun, and pointer status are meaningful. If data is being sourced by the RTDMA and there is some other periodic trigger mechanism trying to initiate transfers, underrun becomes a critical error. If an underrun happens, a buffer went out of sync. This not only affects the current transfer, but can also affect all future transfers due to the ring buffer. Under such conditions, the underrun needs a soft reset to cleanly recover. Alternately, the software can manually stop the transmitting, reset the buffer pointers, clear the remaining error conditions, and then restart transmission. The software method involves a few steps, while the soft reset is a single action and makes sure of a full reset of the control registers.

Due to the flexibility of the transmit buffer, software can implement a simple ping-pong buffer or randomly load and send from any location of the buffer. If the buffer is used in this manner, error flags and status fields can be ignored without adversely affecting the transmitter capability. Additionally, the CURR_WORD_CNT is also invalid if used in this way. The application can set the buffer pointer manually by writing the 4-bit index to TX_BUF_PTR_LOAD. This forces the transmitter to start picking the data from the indicated location in the buffer.

36.3.2.5 CRC Submodule

The FSI transmitter can supply the CRC to the frame being transmitted through the embedded hardware CRC submodule or by supplying a user-defined value. This is controlled by setting TX_OPER_CTRL_LO.SW_CRC appropriately.

If hardware CRC generation is selected (TX_OPER_CTRL_LO.SW_CRC = 0, the default), the CRC is computed by hardware on the data and user data fields using the CRC polynomial 0x7 ($x^8 + x^2 + x + 1$). The transmitter module automatically computes the CRC on the data fields without user intervention when the frame is transmitted. For more information on how the CRC is generated by the CRC submodule, refer to Section 36.3.7.

If software CRC generation is selected (TX_OPER_CTRL_LO.SW_CRC = 1), the CRC must be computed by software and placed in the TX_USER_CRC register. The next frame to be transmitted uses the value placed in the TX_USER_CRC register in place of the CRC value generated by the hardware.

As the TX_USER_CRC register is software-programmable, the application can use this field as an extra data field for application-specific purposes. If TX_USER_CRC is used in this manner, the CRC detection on the receiver is not valid and must be ignored.

36.3.2.6 Conditions in Which the Transmitter Must Undergo a Soft Reset

Unlike the receiver, there are no detectable errors that require a soft reset. A buffer overrun or underrun interrupt can or cannot require a soft reset to resume proper operation. This determination is up to the application software. Refer to Section 36.3.2.4 for more information on the transmit buffer.

36.3.2.7 Reset

The entire transmitter module and all transmitter registers are reset by SYSRSn. The transmitter core is reset by SYSRSn or by writing a 1 to TX_MAIN_CTRL.CORE_RST.

A module reset causes the registers to be reset to the default state.

36.3.3 FSI Receiver Module

The receiver module interfaces to the FSI clock (RXCLK), and data lines (RXD0 and RXD1) after the data lines pass through an optional programmable delay line. The receiver core handles the data framing, CRC computation, and frame-related error checking. The receiver bit clock and state machine are run by the RXCLK input, which is asynchronous to the device system clock.

The receiver control registers allow the CPU to program, control, and monitor the operation of the FSI receiver. The receive data buffer is accessible by the CPU and the RTDMA.

The receiver core has the following features:

- 16-word data buffer
- Multiple supported frame types
- Ping frame watchdog
- Frame watchdog
- CRC calculation and comparison in hardware
- ECC detection
- Programmable delay line control on incoming signals
- RTDMA support
- FSI-SPI compatibility mode

Figure 36-5 provides a high-level overview of the internal modules present in the FSI receiver. Figure 36-6 shows a view of the FSI receiver core submodule. Not all data paths and internal connections are shown.

The following sections describe the various aspects of the FSI receiver module.

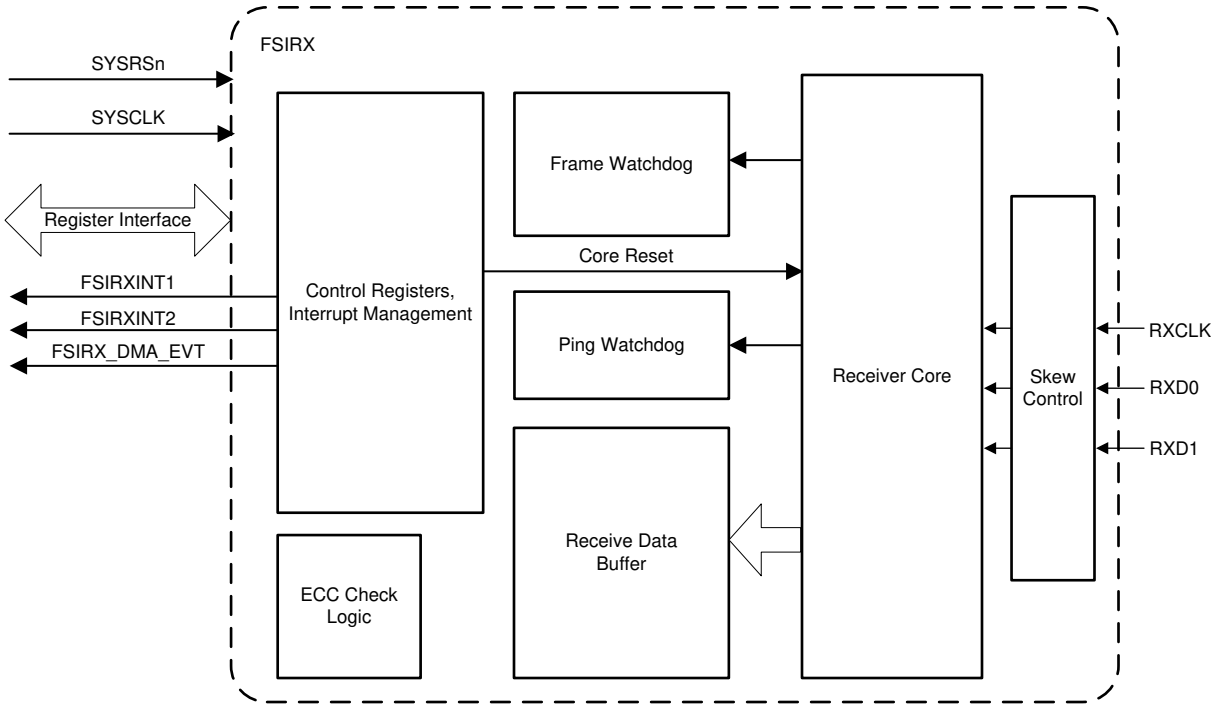


Figure 36-5. FSI Receiver Block Diagram

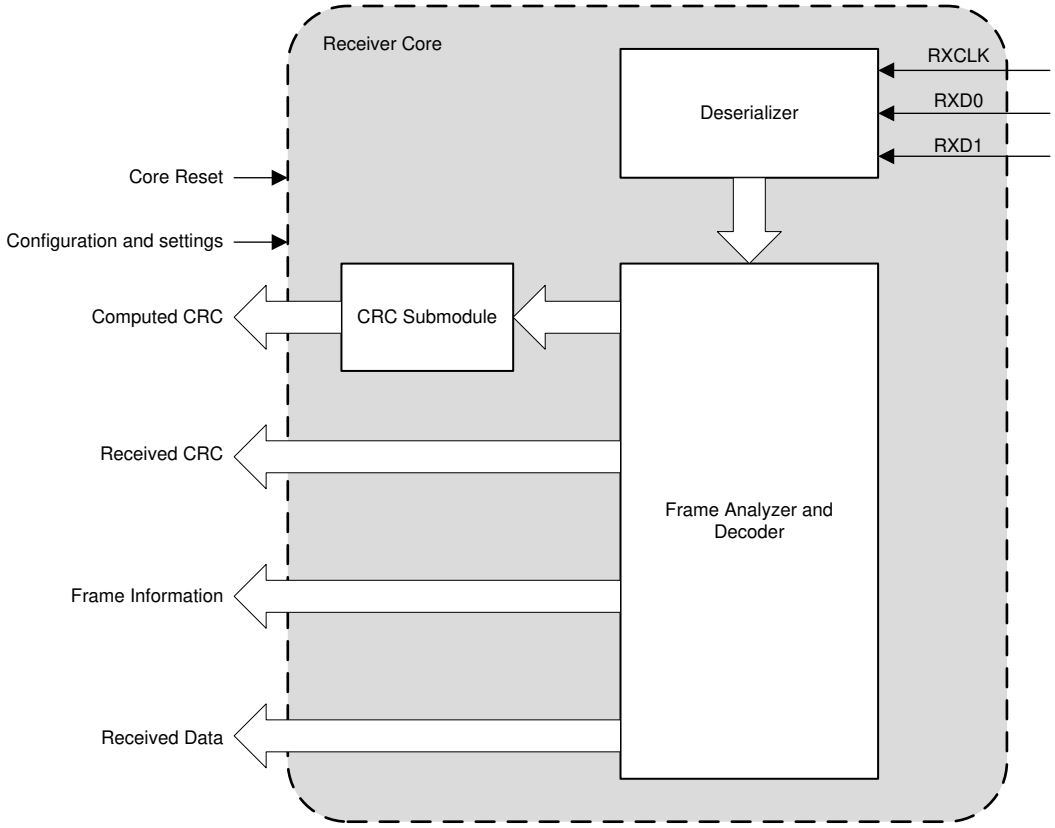


Figure 36-6. FSI Receiver Core Block Diagram

36.3.3.1 Initialization

On the first initialization or after a module reset following any frame error, the receiver module asserts and releases the receiver core reset bit (RX_MAIN_CTRL.CORE_RST) prior to any other initialization. Once the receiver module is initialized, the following steps are executed:

1. If required, assign interrupt sources to the necessary interrupt line.
2. If required, configure the ping watchdog to periodically check for an active link to the transmitter. See Section 36.3.3.4 for configuration details.
3. If required, configure the frame watchdog to make sure that each frame is received within a predetermined window. See Section 36.3.3.5 for configuration details.
4. Initialize the receive buffer pointer by writing to the RX_BUF_PTR_LOAD register. Received data is placed into the buffer starting with the address loaded in this register.
5. Make sure all errors and flags have been cleared from the RX_EVT_STS register.

At this point the receiver is ready to receive any incoming frames. Software can now either poll on the RX_EVT_STS register for various conditions. For example, when the RX_EVT_STS.FRAME_DONE and no other flags are set, the receiver has successfully received a frame without error.

Next, the application configures the various features such as the ping and frame watchdogs, RTDMA, external triggering, and so on. These features are described in subsequent sections. The receiver module is now ready to synchronize with the transmitter then begin reception. The synchronization sequence is described in Section 36.4.1.

36.3.3.2 FSI_RX Clocking

The receiver module registers and control logic are clocked by the device system clock (SYSCLK). The receiver state machine is clocked by the receiver input clock pin (RXCLK).

CAUTION

RXCLK must never be faster than SYSCLK.

36.3.3.3 Receiving Frames

Once the receiver has been properly configured and synchronized, incoming messages are handled as described below. Note that there is no equivalent to a chip-select signal to gate incoming data. Every valid clock edge latches data into the receiver.

The header information of the received frame is placed in the respective register fields.

- RX_FRAME_INFO.FRAME_TYPE contains the received frame type.
- RX_FRAME_TAG_UDATA.FRAME_TAG contains the received frame tag.
- RX_FRAME_TAG_UDATA.USER_DATA contains the received user data.

If any error conditions occur during reception such as a CRC mismatch, frame error, frame timeout, buffer overrun, or ping watchdog timeout, the corresponding flag is set in the RX_EVT_STS register.

Note

If at any point during operation a frame error occurs, the receiver module must be reset and re-synchronized with the transmitter before the next frame can be successfully received. The follow errors are classified as frame errors:

- Type error
- CRC error
- End of frame error

36.3.3.3.1 Receiving Frames with RTDMA

The FSI receiver can continuously receive data and move the data from the receiver buffer with the RTDMA. A RTDMA trigger is generated every time a data frame has been received. This is concurrent with the FRAME_DONE signal that sets the RX_EVT_STS.FRAME_DONE flag. To receive continuous data with the RTDMA, some configurations need to be made on the receiver.

First, set RX_DMA_CTRL.DMA_EVT_EN to 1. This allows the RTDMA trigger to propagate to the RTDMA module. The receiver is now able to trigger a RTDMA event upon the reception of a data frame.

The RTDMA must also be configured properly for the FSI to receive the data. One way for using the receiver to continuously feed the RTDMA is:

- Set up two RTDMA channels to be triggered by the FSI Receiver RTDMA trigger.
- Configure one RTDMA channel to copy data from the receive buffer to a larger data buffer.
- Configure the next RTDMA channel to copy the received frame tag and user data to another data buffer.
- Since the FSI receive buffer is a 16-word circular buffer, make sure the RTDMA channel servicing the data buffer wraps after 16 words are copied.

Unlike the transmitter, there is no requirement to have the RTDMA channel that is handling the data buffer execute before the RTDMA channel handling the received tag and user data.

36.3.3.4 Ping Frame Watchdog

The ping frame watchdog is a hardware-enabled automatic error detection of the connection status to the transmitter. This watchdog monitors the time elapsed between ping frames. If the transmitter has been set up to periodically send out a ping frame, the receiver can be set up to monitor whether this frame has been received within a specified amount of time. If the time between ping frames has exceeded the programmed number of clock cycles, an event is triggered that can generate an interrupt or be monitored by software.

This watchdog has a dedicated counter that is reset and restarted upon the successful reception of a ping frame. The watchdog counter is incremented at the rate of SYSCLK. Optionally, the watchdog can be configured to be reset upon the successful reception of any frame. This option allows the receiver to monitor for any successful frame to indicate that the connection is still alive and the transmitter is still functioning as expected.

To configure the ping frame watchdog for operation:

1. Reset the ping watchdog counter by setting RX_PING_WD_CTRL.PING_WD_RST to 1 and then subsequently clearing the bit to 0.
2. Set RX_OPER_CTRL.PING_WD_RST_MODE to the desired watchdog reset event, set to 0 for ping frames only or set to 1 for any frame.
3. Set RX_PING_WD_REF to the maximum time between frames. Add 10 additional SYSCLK cycles to account for clock synchronization.
4. Enable the ping watchdog by setting RX_PING_WD_CTRL.PING_EN to 1.

The ping watchdog is now enabled and can now monitor for ping frames.

If the RX_PING_WD_CNT value reaches the value programmed in RX_PING_WD_REF, the RX_EVT_STS.PING_WD_TO flag is set. If configured, an interrupt can be generated on this event.

36.3.3.5 Frame Watchdog

The frame watchdog is an additional feature the receiver can use to monitor for any error conditions. This dedicated watchdog monitors the duration for a single frame to be received. The watchdog starts incrementing at the time the receiver detects a proper start of frame condition. If the end of frame condition is not detected within the expected number of SYSCLK cycles, the frame watchdog is triggered that can generate an interrupt or be monitored by software.

This watchdog is automatically started and stopped at the start-of-frame and end-of-frame conditions, respectively. The frame watchdog is connected to SYSCLK.

To configure the frame watchdog for operation:

1. Reset the frame watchdog counter by setting RX_FRAME_WD_CTRL. FRAME_WD_CNT_RST to 1 and then subsequently clearing the bit to 0.
2. Set RX_FRAME_WD_REF.FRAME_WD_REF to the maximum number of SYSCLK cycles expected to be in the longest frame that can be received. Add an additional 10 SYSCLK cycles to account for clock synchronization.
3. Enable the frame watchdog by setting RX_FRAME_WD_CTRL.FRAME_WD_CNT_EN to 1.

The frame watchdog is now enabled and can detect a failed frame.

If the RX_FRAME_WD_CNT reaches the value programmed in RX_FRAME_WD_REF, the RX_EVT_STS.FRAME_WD_TO flag is set. If enabled, an interrupt can be generated on this event.

If the frame watchdog interrupt ever occurs, the receiver core is in an invalid state to receive a new transmission. The only way to recover from a frame watchdog time out is to undergo a soft reset, and subsequently resynchronizing with the transmitter.

36.3.3.6 Delay Line Control

The receiver module has a programmable delay line on each of the external signal inputs: RXCLK, RXD0, and RXD1. The delay elements introduce delays on the respective lines. This is to facilitate adjustment for signal delays introduced by system level components such as signal buffers, ferrite beads, isolators, and so on, or board delays such as uneven trace lengths, long cable length, and so on. The length of the delay is controlled by setting the RX_DLY_LINE_CTRL register values for each line. By default, no delay is introduced by the delay line elements. The delay values must only be adjusted while the FSIRX is held in soft reset, making sure that there are no active transmissions during this process. Figure 36-7 shows a representation of the delay line circuitry for the input signals. The implementation for RXCLK, RXD0, and RXD1 are replicas of this diagram. All circuits behave similarly.

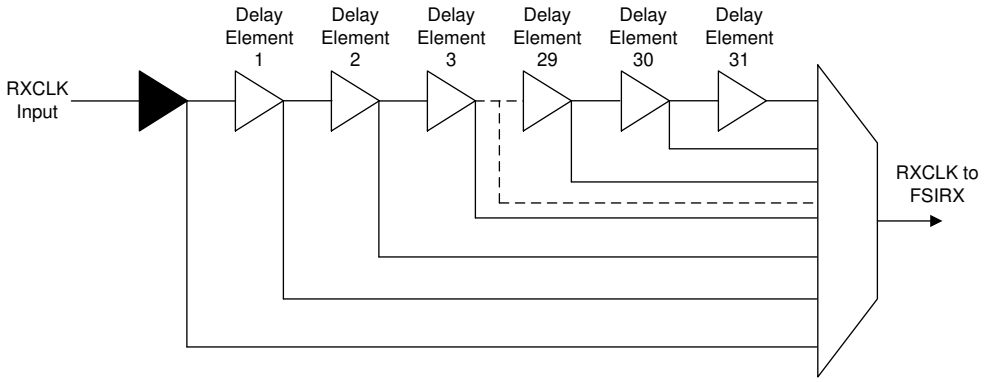


Figure 36-7. Delay Line Control Circuit

For more information on skew compensation, refer to *Fast Serial Interface (FSI) Skew Compensation*.

The FSITX module also has a delay line control circuitry that is placed before the FSITX signals (TXCLK, TXD0, and TXD1) are sent to the TDM signal selection mux (controlled by the SEL_TDM_PATH signal). The TX_DLY_LINE_CTRL register determines the length of the delay for each output line.

36.3.3.7 Buffer Management

The FSI receiver has a 16-word buffer that the data is copied to when the data has been received. This buffer is implemented as a circular buffer, not a FIFO, so some care must be taken to properly interpret buffer overrun and underrun as well as the RX_BUF_PTR_STS register. These flags and pointers work under the assumption that the software or RTDMA is using the buffer as a circular buffer. If the receiver state machine enters into an erroneous state, there is no way for software to cleanly handle this because there is no specified receive clock. For the receiver to detect a clean resynchronization, the state machine needs to be operational and not in the error state. The only way to recover from the error state is to reset the entire receiver module. For overrun and underrun, the receiver can no longer verify that values in the buffer are valid. As such, the best way to recover is to reset the FSI and resynchronize with the transmitter.

Due to the flexibility of the receive buffer, it is possible for software to implement a simple ping-pong buffer, or to randomly receive and read from any location of the buffer. If the buffer is used in this manner, these flags and status fields can be ignored without adversely affecting the receiver capability. Additionally, the CURR_WORD_CNT is also invalid if used in this way. The application can set the buffer pointer manually by writing the 4-bit index to RX_BUF_PTR_LOAD. This forces the receiver to start storing the received data starting at the indicated location in the buffer.

36.3.3.8 CRC Submodule

The receive module automatically calculates the CRC on the incoming data. The received CRC value is placed into RX_CRC_INFO.RX_CRC. The CRC value calculated by hardware on the received data is placed into RX_CRC_INFO.CALC_CRC. These values are compared by hardware and RX_EVT_STS.CRC_ERROR is set if there is a mismatch. The receiver can generate an interrupt based on RX_EVT_STS.CRC_ERROR if enabled.

Since the CRC is only used in data frames, the values found in RX_CRC_INFO.RX_CRC and RX_CRC_INFO.CALC_CRC are undefined during ping and error frames.

For more information on how the CRC is calculated, refer to Section 36.3.7.

If the transmitting module is sending a software-defined CRC value (FSITX.TX_OPER_CTRL_LO.SW_CRC = 1), the receiver module triggers a CRC error event if the received value does not match the hardware-calculated value. As this is an application-level decision, the FSIRX can safely disregard the CRC error event. Application software needs to calculate and verify the incoming CRC using the same custom algorithm used on the transmitter and act appropriately.

The CRC field can also be used as an application-specific value, not a CRC. The application can use the RX_CRC_INFO.RX_CRC as required. All CRC errors and flags can be ignored in this situation.

36.3.3.9 Using the Zero Bits of the Receiver Tag Registers

The receiver tag registers (receiver frame tag and user data (RX_FRAME_TAG_UDATA) register and receiver ping tag (RX_PING_TAG) register)) have the least-significant bit set to 0. The actual received tag is in the bit positions 4:1. The reason for this is to facilitate user software to create a table of functions that can be called depending on the tag value. A function pointer needs a 32-bit storage space and, hence, each successive pointer is offset by 2. If the first pointer is at address x, then the second pointer is at address x + 2, the third at address x + 4, and so on. By keeping the LSB to 0, the five bits of the tag register (bits 4:0) can now be directly used as an index into a table of function pointers.

Note

C29 devices are 8-bit addressable, so each 32-bit pointer requires four addresses. To utilize the frame tag as a pointer on 8-bit addressable devices, the value in the FRAME_TAG field must be multiplied by 2.

36.3.3.10 Conditions in Which the Receiver Must Undergo a Soft Reset

The receiver receives data on every clock edge. While there are specific patterns that determine the a start of a frame, and denote the end of a frame, these patterns are able to occur at any point during normal operation inside of the frame. If there ever is a point at which the receiver fails to detect a successful frame, the module must be reset to make sure that subsequent frames are received properly.

When any of the following errors occur in a received frame, the receiver can be required to be reset and resynchronized with the transmitter:

- Frame type error
- End of frame error
- Ping frame watchdog timeout
- Frame watchdog timeout
- Receiver in an invalid state due to noisy clock

The receiver core status (RX_VIS_1.RX_CORE_STS) can be monitored to determine if the receiver core has entered into an error state requiring a soft reset to resume communication. Incorrect frame type and end of frame errors always cause this bit to become set. A soft reset is required in these cases. A frame watchdog timeout always requires a reset due to the fact that the receiver state machine is still expecting more information when the watchdog timed out. RX_CORE_STS can be used to determine if a noise event was the cause of the failed frame. The ping frame watchdog also does not cause RX_CORE_STS to be set. Similar to the frame watchdog, a corrupt receiver can not be the reason for the ping frame to have timed out. The transmitter can

have gone offline and never sent a ping frame. Alternately, during idle time, a noise event can have occurred, thereby putting the receiver into a corrupt state. As the receiver is able to detect this during the ping frame watchdog timeout interrupt handler, this type of event is not lost and the application can act appropriately.

As the receiver is clocked by RXCLK, not SYSCLK, a noisy clock or data line can cause some internal design constraints to be violated, putting the receiver core logic into undefined states. Make sure that the clock and data lines satisfy the Electrical Characteristics and timing requirements of the FSI module found in the device data sheet. Failure to do so can cause the receiver state machine to go into an unrecoverable error state. The receiver can only be recovered by undergoing a soft reset. To determine the state of the receiver core after an unexpected frame error, the application must check the receiver core status bit.

In addition to the above errors, buffer overrun or underrun can warrant a soft reset to resynchronize with the local application software. Refer to Section 36.3.3.8 for more information on the receive buffers. The requirement of resetting the receiver due to overrun or underrun is up to the application.

After the receiver has been placed into soft reset, the application must notify the other device's transmitter to begin a new synchronization phase. The simplest way to achieve this is through a ping or error frame sent with a designated tag. If the application is not using the FSITX on the device with the detected error, some other method must be established. The other device must stop transmitting and begin a new synchronization phase.

36.3.3.11 FSI_RX Reset

The receiver module and the registers are reset by SYSRSn. The receiver core is reset by SYSRSn or by writing a 1 to RX_MAIN_CTRL.CORE_RST.

A module reset causes the registers to be reset to the default state. After a module reset, the receiver module must be re-initialized and the data link re-established.

36.3.4 Frame Format

The FSI module transmits and receives information in frames. Each frame contains multiple phases where different information can be found. The number of phases as well as the total length of the frame varies depending on the frame type being transmitted. Frames can be as short as 16-bits long for a ping or error frame or 288-bits long for a 16-word data frame.

In normal transmission mode, there are four preamble clock edges before the start of the frame and four post-frame clock edges (postamble). Data is transmitted on both edges of the clock (double data rate). The basic frame structure is shown in Table 36-4. Each phase of the frame (such as start-of-frame, frame type, and so on) is transmitted with the most-significant bit first. Table 36-4 describes the basic frame structure used by the FSI and adapted according to which frame type is transmitted.

Table 36-4. Basic Frame Structure

Idle State	Preamble	Start of Frame	Frame Type	User Data	Data Words	CRC Byte	Frame Tag	End of Frame	Postamble	Idle State
	1111	1001	4 bits	8 bits	1-16 words	8 bits	4 bits	0110	1111	

The FSI also supports a FSI-SPI compatibility mode. The SPI compatible frame structure is similar to a standard FSI frame, but there are differences. Refer to Section 36.3.9 for more information on how to configure and use the FSI-SPI compatibility mode.

Note

One word of the FSI refers to 16 bits.

The terms “frame” and “packet” can be used interchangeably to describe the signaling format of the FSI.

36.3.4.1 FSI Frame Phases

The different phases of the frame structure are described in detail.

- **Idle State:** During the idle state, the clock and data lines are driven high, the inactive state.
- **Preamble:** The preamble phase contains four clock edges (or two complete clock pulses) with the data signals held in the high state. These clock edges serve to flush the receiver logic and prepare the receiver logic for receiving a new frame. This phase is not present in SPI compatibility mode.
- **Start of Frame:** The start of frame phase contains two clock pulses with four bits, 1001, transmitted on the data lines.
- **Frame Type:** The frame type phase contains two clock pulses with the 4-bit frame type code being transmitted on the data lines. The different frame types are described in detail in Section 36.3.4.2. The transmitter must set the TX_FRAME_CTRL.FRAME_TYPE field before transmitting a frame. The received frame type is stored in the RX_FRAME_INFO.FRAME_TYPE.
- **User Data:** The user data phase contains a fully user-configurable data field. There are no restrictions on how this field is used. This phase is only available in data frames. The user data to be transmitted is set by writing to TX_FRAME_TAG_UDATA.USER_DATA. The received user data is stored in RX_FRAME_TAG_UDATA.USER_DATA.
- **Data:** The data phase contains the data that is being transmitted. The data is pulled from the transmit buffer of the transmitter and is placed in the receive buffer of the receiver. Word 0 is transmitted first. This phase is only present in data frames. Depending on the type of frame transmitted, this can contain anywhere between 1 and 16 words depending on the frame type selected. More information on data frames is found in Section 36.3.4.2.3.
- **CRC Byte:** The CRC byte contains the CRC of the transmitted data. The value present in this phase can be sourced from either hardware or software based on the TX_OPER_CTRL_LO.SW_CRC bit. Refer to the module-specific section of the CRC Submodule for more information on the CRC is generated or used, for the transmitter and receiver modules respectively. The CRC byte is only present in data frames.
- **Frame Tag:** The frame tag contains the 4-bit user-defined frame tag. There are no restrictions on how this field is used in an application. The transmitter supplies this tag into the TX_FRAME_TAG_UDATA.FRAME_TAG bits for data frames. Ping frames use the tag defined in TX_PING_TAG.TAG. The receiver can access the received frame tag in RX_FRAME_TAG_UDATA.FRAME_TAG.
- **End of Frame:** The end of frame contains four clock edges with four bits, 0110, transmitted on the data lines.
- **Postamble:** The postamble contains four additional clock edges with the data lines held in the high state. After the postamble, the clock and data lines are driven high (inactive state). This phase is not present in FSI-SPI compatibility mode.

36.3.4.2 Frame Types

The FSI hardware can generate and handle many predefined frame types. The different frame types can be used by the application to signal different types of events or convey different information to the receiver. The different frame types influence which phases and data fields to include in the transmitted frames.

Table 36-5 provides a short overview of the different frame types used by the FSI. Each frame type is described in more detail in the following subsections.

Table 36-5. Frame Types and the 4-bit Codes

Frame Type	4-bit Frame Code	Description
PING	0000	Used typically for checking line integrity. A ping frame can be sent either by software or automatically by hardware.
ERROR	1111	Used typically during error conditions or any condition where one side wants to signal the other side for attention. However, the user software can use an error frame for any purpose.
DATA_1_WORD	0100	1 word data packet (16 bits of data)
DATA_2_WORD	0101	2 word data packet (32 bits of data)
DATA_4_WORD	0110	4 word data packet (64 bits of data)
DATA_6_WORD	0111	6 word data packet (96 bits of data)
DATA_N_WORD	0011	N(1-16) word data packet where software has programmed the number of the data words in a designated register. Both transmitter and receiver modules must have the same value programmed.
Reserved	0001, 0010, and 1000-1110	Reserved

36.3.4.2.1 Ping Frames

Ping frames are one of the most basic frames that can be generated by the FSI. Table 36-6 shows the structure of the ping frames.

Table 36-6. Ping Frame

Idle State	Preamble	SOF	Frame Type	Frame Tag	EOF	Postamble	Idle State
	1111	1001	0000	xxxx	0110	1111	

The ping frame type is always 0000. The frame tag is defined by the application. Separate frame tags exist for timer and software initiated ping frames. No data or CRC is transmitted in a ping frame.

The main purpose of the ping frame is to periodically send a notification to the receiver to make sure an active connection between the transmitter and receiver. The transmitter and receiver cores implement different features to allow the ping frame to operate as a line break detect feature.

On the transmitter, the ping frame is the only frame that can be set up and transmitted without any further software or RTDMA intervention. Ping frames can be transmitted by any (or all) of the three sources: automatic ping timer, software, or external triggers. See Section 36.3.2.3.3 for information on how the transmitter configures and sends the ping frames.

The receiver has a ping watchdog that can detect if a ping frame has not been received in a predetermined window. This allows the receiver to know if the connection between the receiver and the transmitter has been broken. See Section 36.3.3.4 for information on how the receiver handles ping frames.

36.3.4.2.2 Error Frames

Error frames are similar to ping frames in that there are no data fields transmitted. Despite the naming of this frame as an “error frame,” the usage of it is up to the application, as no restrictions are placed on how and when this type of frame is transmitted. Table 36-7 shows the structure of an error frame.

Table 36-7. Error Frame

Idle State	Preamble	SOF	Frame Type	Frame Tag	EOF	Postamble	Idle State
	1111	1001	1111	xxxx	0110	1111	

The structure of the error frame is the same as a ping frame. No data or CRC values are transmitted. The frame type is 1111 for all error frames, and the frame tag is defined by software in the TX_FRAME_TAG_UDATA register.

The receiver can detect if an error frame has been received based on the frame type field. Because of this, the receiver can read the incoming frame tag from the RX_FRAME_TAG_UDATA register and act on up to 16 different conditions.

36.3.4.2.3 Data Frames

Data frames are the most complex frames. As the name indicates, these frames are used to transfer data. Table 36-8 shows the general structure of data frames.

Table 36-8. Data Frame

Idle State	Preamble	SOF	Frame Type	User Data	Data Words	CRC Byte	Frame Tag	EOF	Postamble	Idle State
	1111	1001	0xxx	xxxx xxxx	1-16 words	xxxx xxxx	xxxx	0110	1111	

The frame type field reflects the 4-bit code of the frame type. A list of frame types can be seen in Table 36-5. The number of the data words transmitted is determined by the frame type chosen.

There are four fixed-length data frames supported by the frame type: 1 word, 2 words, 4 words, and 6 words.

Additionally, there is a user-defined data length frame type where the number of data words is fixed by software. Anywhere from 1 to 16 words can be transmitted in this frame type. This length must be configured in the N_WORDS field of the transmitter’s TX_FRAME_CTRL register and receiver’s RX_OPER_CTRL register.

36.3.4.3 Multi-Lane Transmission

The FSI is capable of transmitting and receiving data on two parallel data lines. When enabled, data bits are split between the data lines while the start of frame, frame type, frame tag, and end of frame fields are identical and complete on each line. The user data, data, and CRC fields are split between the data lines. Starting with the most-significant bit, the odd-numbered bits appear on D0 and even-numbered bits appear on D1.

In the following example, assume the following:

8-bit user data: u7u6u5u4u3u2u1u0

16-bit data: d15d14d13d12…d1d0

8-bit CRC: c7c6c5c4c3c2c1c0

Table 36-9. Multi-Lane Frame Format

Idle State	Preamble	SOF	Frame Type	User Data	Data Words	CRC Byte	Frame Tag	EOF	Postamble	Idle State
TXD0	1111	1001	0011	$u^7u^5u^3u^1$	$d^{15}d^{13}…d^1$	$c^7c^5c^3c^1$	xxxx	0110	1111	
TXD1	1111	1001	0011	$u^6u^4u^2u^0$	$d^{14}d^{12}…d^0$	$c^6c^4c^2c^0$	xxxx	0110	1111	

36.3.5 Flush Sequence

Every time there is a soft reset of the receiver, the receiver requires a flush sequence from the transmitter before the receiver can receive and decode frames. The receiver core has an asynchronous reset mechanism that allows the receive module to be reset even in the absence of the receive clocks. However, due to the design, this reset is released synchronous to the receive clock (RXCLK). Thus, the receiver requires five full clock pulses to be able to come out of reset. Sending the flush pattern makes sure that these clock edges are received and any subsequent frames sent to the receiver are correctly interpreted.

The flush sequence consists of a single toggle on both of the data lines as well as five consecutive pulses on the clock line.

If the FSI receiver is receiving data from a standard SPI, a data word of 0xFFFF from the SPI has the same effect as a flush sequence.

Figure 36-8 shows a sample plot of the flush sequence.

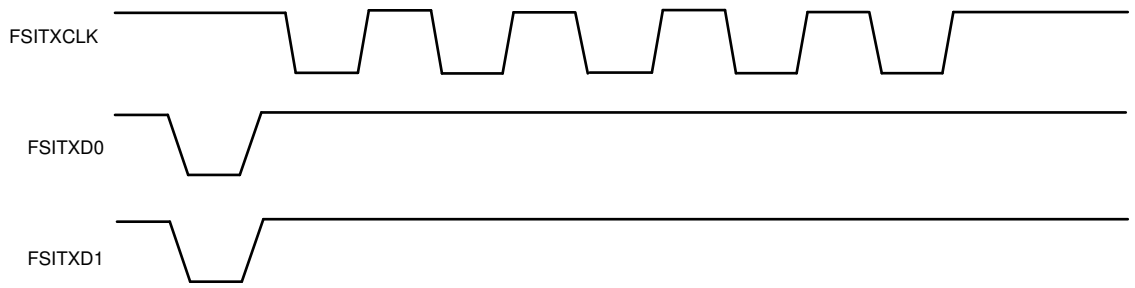


Figure 36-8. Flush Sequence Signals

36.3.6 Internal Loopback

The transmitter and receiver cores can be connected together internally to allow for development and debug. This is achieved by setting RX_MAIN_CTRL.INT_LOOPBACK to 1. Internal loopback routes the signals from the corresponding transmitter to the appropriate receiver pin. No configuration needs to be done in the transmitter. Figure 36-9 shows the signal connections with internal loopback.

In this device, there are four FSI transmitter cores (A, B, C, and D), and four receiver cores (A, B, C, and D). When using loopback mode, each of the FSI transmitter cores can be used in the loopback mode with two FSI receiver cores. Refer to Table 36-10 for specific loopback connections between FSI transmitter and Receiver cores.

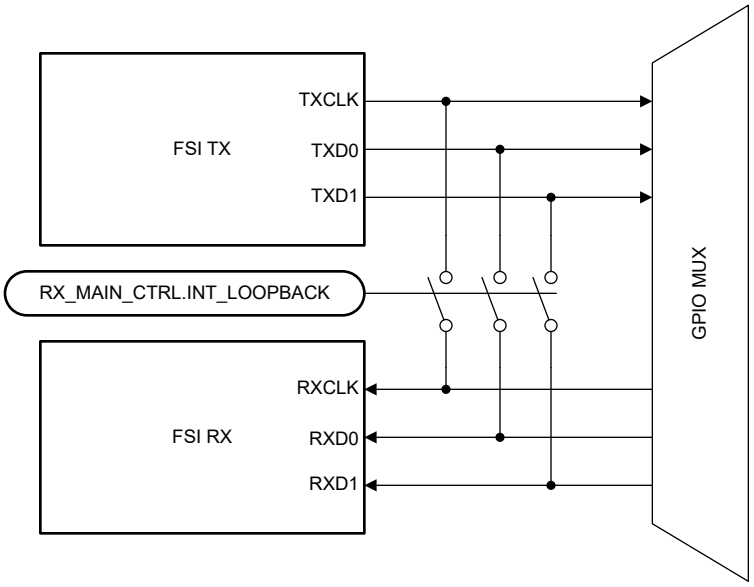


Figure 36-9. FSI with Internal Loopback

Table 36-10. Loopback Connections

TX Module	RX Modules
FSIA TX	FSIA RX, FSIC RX
FSIB TX	FSIB RX, FSID RX
FSIC TX	FSIC RX, FSIA RX
FSID TX	FSID RX, FSIB RX

36.3.7 CRC Generation

The FSI uses CRC-8 with the polynomial 0x07 for the internal hardware CRC generation. This polynomial is also represented as x^8+x^2+x+1.

For example, for a 2-word data packet the following calculation occurs:

Data-1 = 0x4433

Data-0 = 0x2211

User Data = 0xAA

The CRC is computed with the bytes being taken in the following order (first to last):

0xAA – Byte 0, User Data

0x11 – Byte 1, Data-0, Least-significant byte

0x22 – Byte 2, Data-0, Most-significant byte

0x33 – Byte 3, Data-1, Least-significant byte

0x44 – Byte 4, Data-1, Most-significant byte

36.3.8 ECC Module

The FSI module comes with a 16-bit or 32-bit ECC computation module in both the transmitter and receiver. Use of this module is optional.

Note that the ECC is independent and unrelated to the hardware CRC computation module present in both the transmitter and receiver cores.

The following example shows a scenario in which the application requires ECC be calculated and transmitted on a 2-word data frame.

In the FSITX module:

1. Configure the ECC module for 32-bit data by setting TX_OPER_CTRL_HI.ECC_SEL to 1.
2. Write the data to the TX_ECC_DATA register as well as the transmit buffer.
3. Read TX_ECC_VAL Register. This register contains the 8-bit ECC value calculated on the data.
4. Copy the 8-bit data from TX_ECC_VAL to TX_FRAME_TAG_UDATA.USER_DATA.
5. Set the Start Condition to begin the transmission.

The reverse process is followed on the FSIRX module. Once the data frame is received, user software can do the following:

1. Copy the data from the receive buffer to the RX_ECC_DATA register.
2. Copy the received user data that contains the transmitted ECC value from RX_FRAME_TAG_UDATA.USER_DATA to the RX_ECC_VAL register.
3. Read the RX_ECC_LOG register. This contains the result of the ECC computation using the RX_ECC_DATA and RX_ECC_VAL registers.
   a. If no ECC errors were detected, RX_ECC_LOG is 0. The correct data is available in RX_ECC_SEC_DATA.
   b. If a single bit error was detected, RX_ECC_LOG.SBE is 1. The autocorrected data is available in RX_ECC_SEC_DATA.
   c. If multiple bit errors occurred, RX_ECC_LOG.MBE is 1. The data in RX_ECC_SEC_DATA is invalid and must not be used.

Using a 2-word data frame plus using the user data for the ECC is one possible implementation for ECC detection. Another option is to use a larger data frame a allocate one of the data words to be the ECC value.

36.3.9 FSI-SPI Compatibility Mode

The FSI supports a SPI compatibility mode. While the FSI can communicate with a standard SPI module, the FSI supports a limited configuration. The features of this compatibility mode are:

- Data transmits on rising edge and receive on falling edge of the clock.
- Only 16-bit word size is supported.
- TXD1 is driven like an active-low, chip-select signal. The signal is low for the duration for the full frame transmission.
- No receiver chip-select input is required. RXD1 is not used. Data is shifted into the receiver on every active clock edge.
- No preamble or postamble clocks are transmitted. All signals return to the IDLE state after the frame phase is finished.
- It is not possible to transmit in the SPI peripheral configuration because the FSI TXCLK cannot take an external clock source.

Table 36-11 lists the frame structure of the FSI-SPI compatibility mode. Each frame phase is present in this mode. If the FSI is transmitting to a standard SPI module, the SPI must decode the frame structure. Similarly, if the FSI is configured as a SPI peripheral, the standard SPI must encode the transmission to be sent.

Table 36-11. FSI-SPI Compatibility Frame Structure

Idle State	Start of Frame	Frame Type	User Data	Data Words	CRC byte(1)	Frame Tag	End of Frame	Idle State
	1001	4 bits	8 bits	1-16 words	8 bits	4 bits	0110	

(1) The CRC byte is present only in data frames.

Because of the requirement that the standard SPI module encodes the various frame data, this limits the type of modules that can be connected to the FSI in SPI mode. The paired SPI module must have enough functionality to encode and decode the frames.

If the FSI is transmitted to a standard 16-bit SPI, the data is arranged in the following manner. The example provided in Table 36-12 assumes a DATA_2_WORD frame has been sent.

Table 36-12. Contents of Data Received by a Standard SPI

SPI Data	Data Contents
SPI word 0	1001, 0100, 8-bit User Data
SPI word 1	Data word 1
SPI word 2	Data word 2
SPI word 3	8-bit CRC, 4-bit Frame Tag, 0110

36.3.9.1 Available SPI Modes

There are a few wiring schemes available for the FSI to use when communicating with an SPI module.

36.3.9.1.1 FSITX as SPI Controller, Transmit Only

The FSITX can operate as an independent SPI controller module. In this condition, TXCLK is connected to SPICLK, TXD0 is connected to SPIPICO, and TXD1 is connected to $\overline{\text{SPIPTE}}$, the chip select.

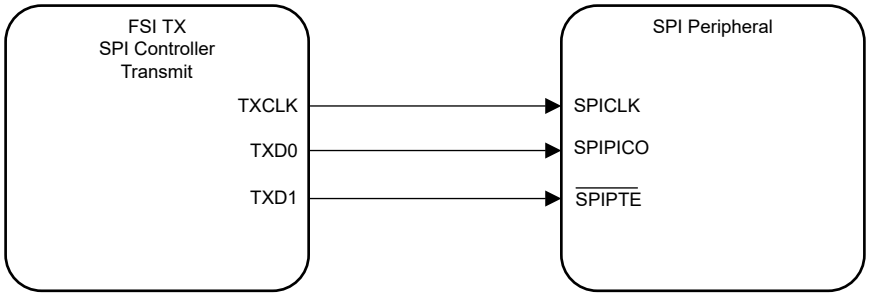


Figure 36-10. FSITX as SPI Controller, Transmit Only

When the FSI is an SPI transmitter, the application has the ability to check for frame errors, line breaks, CRC errors, and ECC checks on data. These are all encoded by hardware in every FSI frame. The SPI receiver requires some software to act upon this information.

Table 36-13. FSI as Controller Transmitter, SPI as Peripheral Receiver

Capability	Availability	Comment
Framing checks on the data frames	Yes	Can be implemented in software on the SPI receiver.
Ability to detect line breaks	Yes	Can be implemented in software on the SPI receiver but requires additional software overhead such as a timer or watchdog.
CRC check	Yes	Can be implemented in software on the SPI receiver. For devices that have VCRC , this is more efficient.
ECC on data	Yes	Can be implemented in software on the SPI receiver
Detection of abruptly terminated frames	No	
Double edge data rate	No	
Recovery from glitches on signal lines between frames	No	
Skew adjustment on signal lines	No	

36.3.9.1.1.1 Initialization

To configure the FSITX module to be an SPI controller for transmit only, proceed through the standard FSITX initialization procedure. Before releasing the FSITX from reset, set TX_OPER_CTRL_LO.SPI_MODE to 1. This enables the SPI clocking scheme and signaling structure.

36.3.9.1.1.2 Operation

The operation of the FSITX module in FSI-SPI Compatibility mode is the same as if the module is in standard FSI mode. The application can utilize the frame timer, ping frames, external frame triggers, and so on. Refer to Section 36.3.2 for more information on each of these features.

36.3.9.1.2 FSIRX as SPI Peripheral, Receive Only

The FSIRX can operate as an independent SPI peripheral module. In this usage, RXCLK is connected to SPICLK and RXD0 is connected to SPIPICO. RXD1 is unused. There is no requirement for a chip select signal to be used when connected to the FSIRX. This is because the FSIRX responds to any incoming clock edge. If there is any noise or unwanted clock transitions, a flush sequence is required to resynchronize the FSIRX module with the controller.

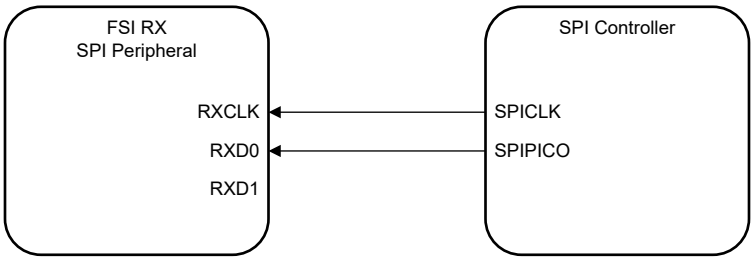


Figure 36-11. FSIRX as SPI Peripheral, Receive Only

When the FSI is an SPI receiver communicating with an SPI transmitter, the application has the ability to detect frame errors, line breaks, CRC errors, ECC checks on data, as well as abruptly terminated frames. Note that the FSI can handle all of this in hardware, but the SPI transmitter must encode the information into the data to be transmitted.

Table 36-14. SPI as Controller Transmitter, FSI as Peripheral Receiver

Capability	Availability	Comment
Framing checks on the data frames	Yes	Standard on FSI
Ability to detect line breaks	Yes	Can be implemented in software on the SPI transmitter but requires the use of a timer or watchdog in the transmitting SPI device.
CRC check	Yes	Can be implemented in software on the SPI transmitter.
ECC on data	Yes	Can be implemented in software on the SPI transmitter.
Detection of abruptly terminated frames	Yes	This is accomplished with the FSI setting up the frame watchdog counter.
Double edge data rate	No	
Recovery from glitches on signal lines between frames	Yes	Whenever glitches occur on either the clock or data lines in between transmissions, the initial flush pattern of a frame discards the effects of these glitches and causes the receiver to resynchronize when the real “start-of-frame” pattern is seen. So, the ability to reject glitches in between frames is very high.
Skew adjustment on signal lines	Yes	The FSI receiver has the ability to add delays to the incoming signal lines.

36.3.9.1.2.1 Initialization

To configure the FSIRX module to be an SPI peripheral for receiving only, proceed through the standard FSIRX initialization procedure. Before releasing the FSIRX from reset, set RX_OPER_CTRL.SPI_MODE to 1. This enables the SPI clocking scheme and signaling structure.

36.3.9.1.2.2 Operation

The operation of the FSIRX module in FSI-SPI compatibility mode is the same as if the module is in standard FSI mode. The application can utilize the Frame and Ping Watchdogs, CRC and ECC checks, and so on. Refer to Section 36.3.3 for more information on each of these features.

36.3.9.1.3 FSITX and FSIRX Emulating a Full Duplex SPI Controller

In this configuration, the FSITX is the controller clock. The FSITX module drives TXCLK (SPICLK), TXD0 (SPIPICO), and TXD1 (SPISTE/chip select) to the SPI peripheral. The SPIPOCI signal is connected back to the RXD0 signal. RXCLK can be applied either using the internal SPI pairing feature or externally wired, depending on the application requirements. Since the FSITX and RX modules are independent, the FSIRX can also be thought of as an additional SPI peripheral. Some software logic is required for the FSI to emulate an SPI controller fully.

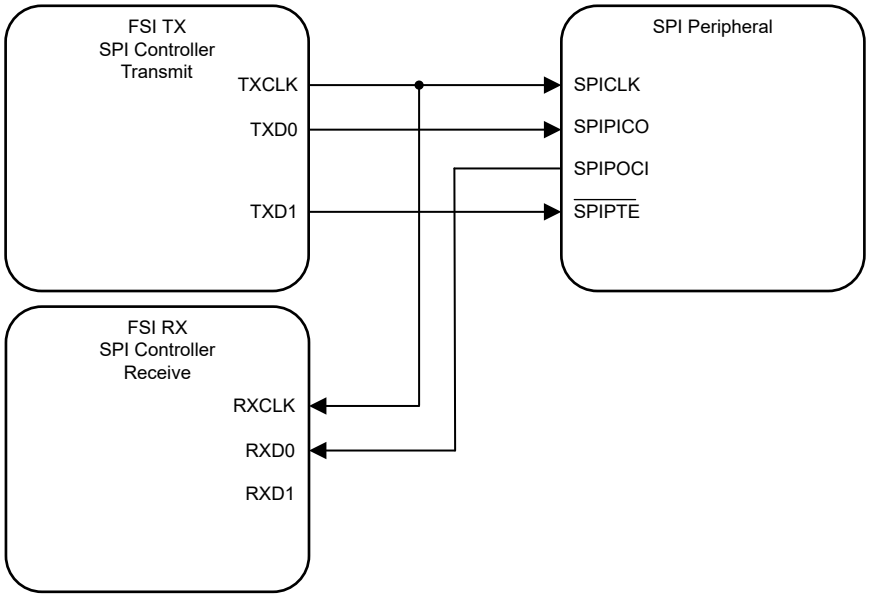


Figure 36-12. FSITX and FSIRX as SPI Controller, Full Duplex

36.3.9.1.3.1 Initialization

To configure both FSITX and RX modules for full duplex SPI controller operation, follow the initialization instructions for each module described in the preceding sections. Both FSITX and RX modules must set the respective SPI_MODE bits. This enables the SPI clocking scheme and signaling structures.

If internal clock loopback is desired, the FSIRX module must also set RX_MAIN_CTRL.SPI_PAIRING to 1. This internally connects TXCLK to RXCLK. If using internal clock loopback, the GPIO used for RXCLK can be reallocated to other application requirements.

If the application requires an external clock loopback, make sure that TXCLK is connected to RXCLK. This is required if the SPI peripheral is across an isolation barrier and there is latency between TXCLK being launched and SPIPOCI data being received on RXD0.

36.3.9.1.3.2 Operation

In this mode of operation, some higher level software must be written to emulate a full SPI controller module. There is no path for the transmit module to determine what the receive module received. Both the TX and RX modules are still able to utilize the various other features available, such as the ping frame timer, ping frame and frame watchdogs, CRC and ECC error checkers, and so on. The procedure for configuring these features is described elsewhere in this document.

36.4 FSI Programing Guide

This section describes various operational sequences and features for the FSI.

36.4.1 Establishing the Communication Link

Once the transmitter and receiver modules have been configured, some synchronization must occur before the modules exchange data. Since the receiver accepts data on any clock transition, the receiver core logic must be flushed to properly interpret the start of a new, valid frame. This is especially true when the FSI modules reside on separate devices and are possibly isolated.

The following example provides a suggested approach for establishing a clean communication link on two separate devices that power up in an arbitrary order. Note that this is only a sample synchronization. Depending on application requirements, a different approach can be followed. The single, most important aspect of synchronization is to make sure that the receiver is properly flushed and ready to receive a complete frame without error. How to achieve this is up to the application.

Figure 36-13 shows the connection of the devices in this example. While there is no true concept of a main device or a remote device node in the FSI protocol, the example uses this nomenclature as a simple way to describe the data flow.

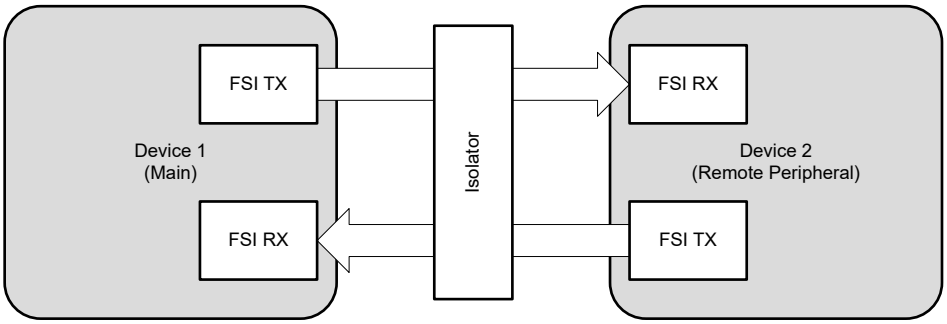


Figure 36-13. Point to Point Connection

Device 1 is the main node; it is the driver of the initialization sequence. Device 2 is the remote node; it responds to the main device commands. In this example, as well as in a real world use-case, neither the main device nor the remote device knows precisely when the other is ready to receive communication.

Sample sequences for both the main device and remote device are provided in the following subsections.

36.4.1.1 Establishing the Communication Link from the Main Device

The following sequence is an example of how the main device node establishes the communication link with the remote device without external signals outside of the standard communication link.

1. Assert the core reset to both the FSITX and FSIRX modules, and then deassert the resets.
2. Configure the transmitter and receiver for desired operation.
3. Set up the receiver interrupts to detect an incoming transmission.
4. Begin the ping loop:
   - Send the flush sequence.
   - Send a ping frame with the frame tag 0000.
   - Wait for some time. (determined by application)
   - If the FSIRX has received a valid ping frame, continue; else, iterate the loop again.
   - If the received ping frame tag was 0001, continue; else, iterate the loop again.
5. Send a ping frame with the frame tag 0001.

At this point, both the main transmit and receive channels have successfully received a frame from the remote counterparts. The link has been established and standard application communication can begin.

36.4.1.2 Establishing the Communication Link from the Remote Device

The following sequence is an example of how the remote device node establishes the communication link with the main device without external signals outside of the standard communication link.

1. Apply the core reset to both the FSITX and FSIRX modules, and then release the reset.
2. Configure the transmitter and receiver for desired operation.
3. Set up the receiver interrupts to detect an incoming transmission.
4. Wait for a receiver interrupt.
5. If the FSIRX has received a valid ping frame, continue; else, return to step 4.
6. If the received frame tag was 0000, continue; else, discard the transmission and return to step 4.
7. Send the flush sequence.
8. Send a ping frame with the frame tag 0001.
9. Wait for a receiver interrupt.
10. If the FSIRX has received a valid ping frame, continue; else, return to step 4.
11. If the received ping frame tag was 0001, continue; else, if the received frame tag was 0000, return to step 9. This can happen if a second ping frame was already in transit before receiving the remote device response in step 8.

At this point, both the transmit and receive modules have successfully received ping frames from the main counterparts. The link has been established and regular communication can now proceed. The application can configure periodic ping frames from the transmitter, initialize the receiver ping and frame watchdogs, and begin the communication required by the application.

36.4.2 Register Protection

Both the FSITX and FSIRX modules contain control registers that have embedded write protection. This is accomplished through EALLOW, register keys, and a main register lock. These protections make sure that no spurious writes or unintentional modifications to these registers are accepted. For the list of registers with write protections available and the register and bit descriptions, refer to the *FSI Registers* section..

EALLOW Protection

EALLOW is a device-level register protection. For those registers with EALLOW protection, the EALLOW bit is set before modifying the register. The application then clears the EALLOW bit to re-enable the write protection when access to EALLOW-protected registers are complete.

Register Key Protection

In addition to EALLOW, some bits in the FSI registers are protected by a key. To write to these bits, the key must be written at the same time. For example, to put the transmitter core into reset, TX_MAIN_CTRL.CORE_RST must be set. To do this, write 0xA501 to TX_MAIN_CTRL, where 0xA500 is the KEY value, and 0x0001 is the CORE_RST value. Refer to the *Registers* section for more information on which registers have write keys added.

Control Register Lock Protection

There also exists a main lock to prevent any modifications to the control registers. There is an independent lock for each FSI module. For the list of registers that are protected by this control register lock, refer to the *Registers* section. The control register lock prevents any writes to the control registers until the lock is released. To set the control register lock, write 0xA501 to RX_LOCK_CTRL and TX_LOCK_CTRL for the receiver and transmitter, respectively.

The control register lock cannot be disabled by the application until a SYSRSn has been asserted. This can occur at the device level, or by writing to the appropriate peripheral soft reset register (DEV_CFG_REGS.SOFTPRESx) for the FSI module. Refer to Section 36.3.2.7 for more information on SYSRSn.

36.4.3 Emulation Mode

There is no specific emulation mode or configuration supported. The FSI cores are always in free running mode. CPU halts do not have any effect on the operation of the FSI. Reads of registers and data buffers by the debugger do not affect any flags or status of the data buffers.

If you want to stop the operation of either FSI module when the debugger halts, the following steps are required:

1. Set the debugger to real-time emulation mode.
2. Mark the FSI interrupt group as a time-critical interrupt. That is, enable the corresponding bit in the DBGIER register.
3. The ISR can check the DSTAT register and to determine if the ISR was called when the debugger was halted.
4. FSI operations can be disabled and the ISR can branch to a debug-specific halt location.

36.5 Software

36.5.1 FSI Registers to Driverlib Functions

Table 36-15. FSI Registers to Driverlib Functions

File	Driverlib Function
TX_MAIN_CTRL	
fsi.c	FSI_resetTxModule
fsi.c	FSI_clearTxModuleReset
fsi.h	FSI_sendTxFlush
fsi.h	FSI_stopTxFlush
TX_CLK_CTRL	
fsi.c	FSI_resetTxModule
fsi.c	FSI_clearTxModuleReset
fsi.h	FSI_enableTxClock
fsi.h	FSI_disableTxClock
fsi.h	FSI_configPrescalar
TX_OPER_CTRL_LO	
fsi.h	FSI_selectTxPLLClock
fsi.h	FSI_setTxDataWidth
fsi.h	FSI_enableTxSPIMode
fsi.h	FSI_disableTxSPIMode
fsi.h	FSI_setTxStartMode
fsi.h	FSI_setTxPingTimeoutMode
fsi.h	FSI_enableTxTDMMode
fsi.h	FSI_disableTxTDMMode
fsi.h	FSI_enableRxTDMMode
fsi.h	FSI_disableRxTDMMode
fsi.h	FSI_enableTxUserCRC
fsi.h	FSI_disableTxUserCRC
TX_OPER_CTRL_HI	
fsi.h	FSI_setTxExtFrameTrigger
fsi.h	FSI_enableTxCRCForceError
fsi.h	FSI_disableTxCRCForceError
fsi.h	FSI_setTxECCComputeWidth
TX_FRAME_CTRL	
fsi.h	FSI_setTxFrameType
fsi.h	FSI_setTxSoftwareFrameSize
fsi.h	FSI_startTxTransmit
TX_FRAME_TAG_UDATA	
fsi.h	FSI_setTxFrameTag
fsi.h	FSI_setTxUserDefinedData
TX_BUF_PTR_LOAD	
fsi.h	FSI_setTxBufferPtr
TX_BUF_PTR_STS	
fsi.h	FSI_getTxBufferPtr
fsi.h	FSI_getTxWordCount
TX_PING_CTRL	

Table 36-15. FSI Registers to Driverlib Functions (continued)

File	Driverlib Function
fsi.c	FSI_resetTxModule
fsi.c	FSI_clearTxModuleReset
fsi.h	FSI_enableTxPingTimer
fsi.h	FSI_disableTxPingTimer
fsi.h	FSI_enableTxExtPingTrigger
fsi.h	FSI_disableTxExtPingTrigger
TX_PING_TAG	
fsi.h	FSI_enableTxPingTimer
fsi.h	FSI_setTxPingTag
TX_PING_TO_REF	
fsi.h	FSI_enableTxPingTimer
TX_PING_TO_CNT	
fsi.h	FSI_getTxCurrentPingTimeoutCounter
TX_INT_CTRL	
fsi.h	FSI_enableTxInterrupt
fsi.h	FSI_disableTxInterrupt
TX_DMA_CTRL	
fsi.h	FSI_enableTxDMAEvent
fsi.h	FSI_disableTxDMAEvent
TX_LOCK_CTRL	
fsi.h	FSI_lockTxCtrl
TX_EVT_STS	
fsi.h	FSI_getTxEventStatus
TX_EVT_CLR	
fsi.h	FSI_clearTxEvents
TX_EVT_FRC	
fsi.h	FSI_forceTxEvents
TX_USER_CRC	
fsi.h	FSI_enableTxUserCRC
TX_ECC_DATA	
fsi.h	FSI_setTxECCdata
TX_ECC_VAL	
fsi.h	FSI_getTxECCValue
TX_DLYLINE_CTRL	
-	
TX_BUF_BASE(i)	
fsi.c	FSI_writeTxBuffer
fsi.h	FSI_getTxBufferAddress
RX_MAIN_CTRL	
fsi.c	FSI_resetRxModule
fsi.c	FSI_clearRxModuleReset
fsi.h	FSI_enableRxInternalLoopback
fsi.h	FSI_disableRxInternalLoopback
fsi.h	FSI_enableRxSPIPairing
fsi.h	FSI_disableRxSPIPairing

Table 36-15. FSI Registers to Driverlib Functions (continued)

File	Driverlib Function
RX_OPER_CTRL	
fsi.h	FSI_setRxDataWidth
fsi.h	FSI_enableRxSPIMode
fsi.h	FSI_disableRxSPIMode
fsi.h	FSI_setRxSoftwareFrameSize
fsi.h	FSI_setRxECCComputeWidth
fsi.h	FSI_setRxPingTimeoutMode
RX_FRAME_INFO	
fsi.h	FSI_getRxFrameType
RX_FRAME_TAG_UDATA	
fsi.h	FSI_getRxFrameTag
fsi.h	FSI_getRxUserDefinedData
RX_DMA_CTRL	
fsi.h	FSI_enableRxDMAEvent
fsi.h	FSI_disableRxDMAEvent
RX_EVT_STS	
fsi.h	FSI_getRxEventStatus
RX_CRC_INFO	
fsi.h	FSI_getRxReceivedCRC
fsi.h	FSI_getRxComputedCRC
RX_EVT_CLR	
fsi.h	FSI_clearRxEvents
RX_EVT_FRC	
fsi.h	FSI_forceRxEvents
RX_BUF_PTR_LOAD	
fsi.h	FSI_setRxBufferPtr
RX_BUF_PTR_STS	
fsi.h	FSI_getRxBufferPtr
fsi.h	FSI_getRxWordCount
RX_FRAME_WD_CTRL	
fsi.c	FSI_resetRxModule
fsi.c	FSI_clearRxModuleReset
fsi.h	FSI_enableRxFrameWatchdog
fsi.h	FSI_disableRxFrameWatchdog
RX_FRAME_WD_REF	
fsi.h	FSI_enableRxFrameWatchdog
RX_FRAME_WD_CNT	
fsi.h	FSI_getRxFrameWatchdogCounter
RX_PING_WD_CTRL	
fsi.c	FSI_resetRxModule
fsi.c	FSI_clearRxModuleReset
fsi.h	FSI_enableRxPingWatchdog
fsi.h	FSI_disableRxPingWatchdog
RX_PING_TAG	
fsi.h	FSI_getRxPingTag

Table 36-15. FSI Registers to Driverlib Functions (continued)

File	Driverlib Function
fsi.h	FSI_setRxPingTagRef
fsi.h	FSI_getRxPingTagRef
fsi.h	FSI_setRxPingTagMask
fsi.h	FSI_getRxPingTagMask
fsi.h	FSI_enableRxPingTagCompare
fsi.h	FSI_disableRxPingTagCompare
fsi.h	FSI_enableRxPingBroadcast
fsi.h	FSI_disableRxPingBroadcast
RX_PING_WD_REF	
fsi.h	FSI_enableRxPingWatchdog
RX_PING_WD_CNT	
fsi.h	FSI_getRxPingWatchdogCounter
RX_INT1_CTRL	
fsi.h	FSI_enableRxInterrupt
fsi.h	FSI_disableRxInterrupt
RX_INT2_CTRL	
fsi.h	FSI_enableRxInterrupt
fsi.h	FSI_disableRxInterrupt
RX_LOCK_CTRL	
fsi.h	FSI_lockRxCtrl
RX_ECC_DATA	
fsi.h	FSI_setRxECCData
RX_ECC_VAL	
fsi.h	FSI_setRxReceivedECCValue
RX_ECC_SEC_DATA	
fsi.h	FSI_getRxECCCorrectedData
RX_ECC_LOG	
fsi.h	FSI_getRxECCLog
RX_FRAME_TAG_CMP	
fsi.h	FSI_setRxFrameTagRef
fsi.h	FSI_getRxFrameTagRef
fsi.h	FSI_setRxFrameTagMask
fsi.h	FSI_getRxFrameTagMask
fsi.h	FSI_enableRxFrameTagCompare
fsi.h	FSI_disableRxFrameTagCompare
fsi.h	FSI_enableRxFrameBroadcast
fsi.h	FSI_disableRxFrameBroadcast
RX_PING_TAG_CMP	
fsi.h	FSI_setRxPingTagRef
fsi.h	FSI_getRxPingTagRef
fsi.h	FSI_setRxPingTagMask
fsi.h	FSI_getRxPingTagMask
fsi.h	FSI_enableRxPingTagCompare
fsi.h	FSI_disableRxPingTagCompare
fsi.h	FSI_enableRxPingBroadcast

Table 36-15. FSI Registers to Driverlib Functions (continued)

File	Driverlib Function
fsi.h	FSI_disableRxPingBroadcast
RX_TRIG_CTRL_0	
-	
RX_TRIG_WIDTH_0	
-	
RX_DLYLINE_CTRL	
fsi.c	FSI_configRxDelayLine
RX_TRIG_CTRL_1	
-	
RX_TRIG_CTRL_2	
-	
RX_TRIG_CTRL_3	
-	
RX_VIS_1	
-	
RX_UDATA_FILTER	
-	
RX_BUF_BASE(i)	
fsi.c	FSI_readRxBuffer
fsi.h	FSI_getRxBufferAddress

36.5.2 FSI Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/fsi

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

36.5.2.1 FSI Loopback:CPU Control - SINGLE_CORE

FILE: fsi_ex1_loopback_cpucontrol.c

Example sets up infinite data frame transfers where trigger happens through *CPU*. Automatic(Hw triggered) Ping frame transmission is also setup along with data.

User can edit some of configuration parameters as per usecase. These are as below. Default values can be referred in code where these globals are defined

- *txUserData* - User data to be sent with Data frame
- *txDataFrameTag* - Frame tag used for Data transfers
- *txPingFrameTag* - Frame tag used for Ping transfers

For any errors during transfers i.e. *error* events such as Frame Overrun, Underrun, Watchdog timeout and CRC/EOF/TYPE errors, execution will stop immediately and status variables can be looked into for more details. Execution will also stop for any mismatch between received data and sent ones and also if transfers takes unusually long time(detected through software counters - txTimeOutCntr and rxTimeOutCntr)

External Connections

- None.

Watch Variables

- *dataFrameCntr* Number of Data frame transfered
- *error* Non zero for transmit/receive data mismatch

36.5.2.2 FSI data transfers upon CPU Timer event - SINGLE_CORE

FILE: fsi_ex2_periodic_frame.c

Example sets up infinite data frame transfers where trigger comes from ISR handling the periodic CPU Timer event. Automatic(Hw triggered) Ping frame transmission is also setup along with data.

CPU Timer0 is chosen for setting up periodic timer events. User can choose any other Timer-1/Timer-2 as well.

Automatic(Hw triggered) Ping frame transmission is also setup along with data.

If there are any comparison failures during transfers or any of error event occurs, execution will stop.

External Connections

- None

Watch Variables

- *dataFrameCntr* Number of Data frame transfered
- *error* Non zero for transmit/receive data mismatch

36.6 FSI Registers

This Section describes the FSI Registers.

36.6.1 FSI Base Address Table

Table 36-16. FSI Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
FSI_TX_REGS	FSITXA_BASE	0x7018_0000	YES	YES	YES	YES	YES	YES	-	YES
FSI_TX_REGS	FSITXB_BASE	0x7018_1000	YES	YES	YES	YES	YES	YES	-	YES
FSI_TX_REGS	FSITXC_BASE	0x7018_2000	YES	YES	YES	YES	YES	YES	-	YES
FSI_TX_REGS	FSITXD_BASE	0x7018_3000	YES	YES	YES	YES	YES	YES	-	YES
FSI_RX_REGS	FSIRXA_BASE	0x7018_8000	YES	YES	YES	YES	YES	YES	-	YES
FSI_RX_REGS	FSIRXB_BASE	0x7018_9000	YES	YES	YES	YES	YES	YES	-	YES
FSI_RX_REGS	FSIRXC_BASE	0x7018_A000	YES	YES	YES	YES	YES	YES	-	YES
FSI_RX_REGS	FSIRXD_BASE	0x7018_B000	YES	YES	YES	YES	YES	YES	-	YES

36.6.2 FSI_TX_REGS Registers

Table 36-17 lists the memory-mapped registers for the FSI_TX_REGS registers. All register offset addresses not listed in Table 36-17 should be considered as reserved locations and the register contents should not be modified.

Table 36-17. FSI_TX_REGS Registers

Offset	Acronym	Register Name	Protection
0h	TX_MAIN_CTRL	Transmit main control register	
4h	TX_CLK_CTRL	Transmit clock control register	and LOCK
8h	TX_OPER_CTRL_LO	Transmit operation control register low	and LOCK
Ah	TX_OPER_CTRL_HI	Transmit operation control register high	and LOCK
Ch	TX_FRAME_CTRL	Transmit frame control register	
Eh	TX_FRAME_TAG_UDATA	Transmit frame tag and user data register	
10h	TX_BUF_PTR_LOAD	Transmit buffer pointer control load register	
12h	TX_BUF_PTR_STS	Transmit buffer pointer control status register	
14h	TX_PING_CTRL	Transmit ping control register	and LOCK
16h	TX_PING_TAG	Transmit ping tag register	
18h	TX_PING_TO_REF	Transmit ping timeout counter reference	and LOCK
1Ch	TX_PING_TO_CNT	Transmit ping timeout current count	
20h	TX_INT_CTRL	Transmit interrupt event control register	and LOCK
22h	TX_DMA_CTRL	Transmit DMA event control register	and LOCK
24h	TX_LOCK_CTRL	Transmit lock control register	and LOCK
28h	TX_EVT_STS	Transmit event and error status flag register	
2Ch	TX_EVT_CLR	Transmit event and error clear register	
2Eh	TX_EVT_FRC	Transmit event and error flag force register	
30h	TX_USER_CRC	Transmit user-defined CRC register	
40h	TX_ECC_DATA	Transmit ECC data register	
44h	TX_ECC_VAL	Transmit ECC value register	
48h	TX_DLYLINE_CTRL	Transmit delay Line control register	and LOCK
80h + formula	TX_BUF_BASE_y	Base address for transmit buffer	

Complex bit access types are encoded to fit into small table cells. Table 36-18 shows the codes that are used for access types in this section.

Table 36-18. FSI_TX_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.

Table 36-18. FSI_TX_REGS Access Type Codes (continued)

Access Type	Code	Description
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

36.6.2.1 TX_MAIN_CTRL Register (Offset = 0h) [Reset = 0000h]

TX_MAIN_CTRL is shown in Figure 36-14 and described in Table 36-19.

Return to the Summary Table.

Transmit main control register

Figure 36-14. TX_MAIN_CTRL Register

15	14	13	12	11	10	9	8
KEY							
W-0h							
7	6	5	4	3	2	1	0
RESERVED						FLUSH	CORE_RST
R-0h						R/W-0h	R/W-0h

Table 36-19. TX_MAIN_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	KEY	W	0h	Write Key In order to write to any bit in this register, 0xA5 must be written to this field at the same time. Otherwise, writes are ignored. The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
7-2	RESERVED	R	0h	Reserved
1	FLUSH	R/W	0h	Flush Operation Start bit This bit will cause the transmitter to initiate a flush pattern of a single toggle on the TXD0 and TXD1 followed by five full cycles of TXCLK. This bit should be written only when the CORE_RST bit is 0 and the clock to the Transmitter core is turned on. 0h (R/W) = Clear this bit. 1h (R/W) = Setting this bit will Initiate flush sequence. To properly execute a flush sequence, Set FLUSH to 1, wait for five TXCLK cycles then clear FLUSH to 0. Note: The KEY field must contain 0xA5 for any write to this bit to take effect. The software must keep this bit set to 1 for at least five TXCLK cycles before setting it back to 0. Reset type: SYSRSn
0	CORE_RST	R/W	0h	Transmitter Main Core Reset bit This bit controls the transmitter main core reset. In order to send any frame, this bit must be cleared. 0h (R/W) = Transmitter core is not in reset and can transmit frames. 1h (R/W) = Transmitter core is held in reset. Note: The KEY field must contatin 0xA5 for any write to this bit to take effect. Reset type: SYSRSn

36.6.2.2 TX_CLK_CTRL Register (Offset = 4h) [Reset = 0000h]

TX_CLK_CTRL is shown in Figure 36-15 and described in Table 36-20.

Return to the Summary Table.

Transmit clock control register

Figure 36-15. TX_CLK_CTRL Register

15	14	13	12	11	10	9	8
RESERVED						PRESCALE_VAL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
PRESCALE_VAL						CLK_EN	CLK_RST
R/W-0h						R/W-0h	R/W-0h

Table 36-20. TX_CLK_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-2	PRESCALE_VAL	R/W	0h	Clock Divider Prescale Value The input clock is divided by this 8-bit value and fed into the transmitter core. This divided clock is the rate at which TXCLK will operate. 0h (R/W) = Reserved 1h (R/W) = Input clock /1 2h (R/W) = Input clock /2 3h (R/W) = Input clock /3 4h (R/W) = Input clock /4 ... FFh (R/W) = Input clock /255 TXCLKIN = Input clock / PRESCALE_VAL In FSI mode: TXCLK = TXCLKIN / 2 In SPI mode: TXCLK = TXCLKIN Reset type: SYSRSn
1	CLK_EN	R/W	0h	Clock Divider Enable bit This bit will enable and disable the input clock divider and start the clock to the transmitter core. 0h (R/W) = The input clock divider is not enabled and the clock is not connected to the transmitter core. 1h (R/W) = The input clock to the transmitter core is being divided by the PRESCALE_VAL and enabled. Reset type: SYSRSn
0	CLK_RST	R/W	0h	Clock Divider Reset bit This bit will reset the clock counter in the clock divider. 0h (R/W) = The clock divider is set based on the value in PRESCALE_VAL. The input clock will be divided by PRESCALE_VAL if CLK_EN is set. 1h (R/W) = The clock divider will be reset to 0 and will stay reset until software writes a 0 to this bit. Reset type: SYSRSn

36.6.2.3 TX_OPER_CTRL_LO Register (Offset = 8h) [Reset = 0000h]

TX_OPER_CTRL_LO is shown in Figure 36-16 and described in Table 36-21.

Return to the Summary Table.

Transmit operation control register low

Figure 36-16. TX_OPER_CTRL_LO Register

15	14	13	12	11	10	9	8
RESERVED					SEL_TDM_IN	TDM_ENABLE	SEL_PLLCLK
R-0h					R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
PING_TO_MODE	SW_CRC	START_MODE			SPI_MODE	DATA_WIDTH	
R/W-0h	R/W-0h	R/W-0h			R/W-0h	R/W-0h	

Table 36-21. TX_OPER_CTRL_LO Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R	0h	Reserved
10	SEL_TDM_IN	R/W	0h	Input TDM port Select bit This bit selects the input port for the transmitter core between the TDM input pins or the RX module. When this bit is '0', the inputs selected for TDM are from the TDM input pins. When this bit is '1', then inputs selected for TDM are from the RX module. Reset type: SYSRSn
9	TDM_ENABLE	R/W	0h	Transmit TDM Mode Enable bit. This bit enables the TDM Mode for multi-remote TDM operation. 0h (R/W) Transmit TDM Mode is not enabled. 1h (R/W) Transmit TDM Mode is enabled. Reset type: SYSRSn
8	SEL_PLLCLK	R/W	0h	Input Clock Select bit This bit selects the input clock source for the transmitter core. 0h (R/W) = SYSCLK is the source of the transmitter clock into the clock prescaler. 1h (R/W) = PLLRAWCLK is the source of the transmitter core clock into the clock prescaler. Reset type: SYSRSn
7	PING_TO_MODE	R/W	0h	Ping Counter Reset Mode Select bit This bit selects when the ping counter will reset. 0h (R/W) = The ping counter will reset and restart only on hardware initiated ping frames, when ping counter has timed out. 1h (R/W) = The ping counter will reset and restart on any software initiated frame as well as a ping counter timeout Reset type: SYSRSn
6	SW_CRC	R/W	0h	CRC Source Select bit This bit selects the source of the CRC value that is transmitted. 0h (R/W) = The transmitted CRC value is computed by hardware. 1h (R/W) = The transmitted CRC value is sourced from the value programmed in the TX_USER_CRC register. Reset type: SYSRSn

Table 36-21. TX_OPER_CTRL_LO Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-3	START_MODE	R/W	0h	Transmission Start Mode Select bit These bits select the method by which a new frame transmission is started. 0h (R/W) = Only a software write to TX_FRAME_CTRL.START initiate a new transmission. 1h (R/W) = The configured external trigger will initiate a new transmission. 2h (R/W) = Either writing to TX_FRAME_CTRL.START or the TX_FRAME_TAG_UDATA register will initiate a new transmission. All other combinations of bits are illegal and reserved for future use. Reset type: SYSRSn
2	SPI_MODE	R/W	0h	SPI Mode Select bit This bit enables and disables SPI compatibility mode. 0h (R/W) = FSI is in normal mode of operation. 1h (R/W) = FSI is operating in SPI compatibility mode. Reset type: SYSRSn
1-0	DATA_WIDTH	R/W	0h	Transmit Data Width Select bits These bits define the number of data lines used by the transmitter. 0h (R/W) = Data will be transmitted on one data line (TXD0) 1h (R/W) = Data will be transmitted on two data lines (TXD0 and TXD1). The format of the data is described in the preceeding chapter. 2h, 3h (R/W) = Reserved Reset type: SYSRSn

36.6.2.4 TX_OPER_CTRL_HI Register (Offset = Ah) [Reset = 0000h]

TX_OPER_CTRL_HI is shown in Figure 36-17 and described in Table 36-22.

Return to the Summary Table.

Transmit operation control register high

Figure 36-17. TX_OPER_CTRL_HI Register

15	14	13	12	11	10	9	8
RESERVED		EXT_TRIG_SEL					
R-0h		R/W-0h					
7	6	5	4	3	2	1	0
EXT_TRIG_SEL	ECC_SEL	FORCE_ERR	RESERVED				
R/W-0h	R/W-0h	R/W-0h	R-0h				

Table 36-22. TX_OPER_CTRL_HI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	RESERVED	R	0h	Reserved
13-7	EXT_TRIG_SEL	R/W	0h	External Trigger Select bit These bits define which of the 128 external inputs will be used as the source for the external input trigger. 00h (R/W) = Trigger 1 is the source. 01h (R/W) = Trigger 2 is the source. 02h (R/W) = Trigger 3 is the source. ... 7Fh (R/W) = Trigger 128 is the source. Reset type: SYSRSn
6	ECC_SEL	R/W	0h	ECC Data Width Select bit This bit selects between 16-bit and 32-bit ECC computation. 0h (R/W) = 32-bit ECC is used. 1h (R/W) = 16-bit ECC is used. Reset type: SYSRSn
5	FORCE_ERR	R/W	0h	Error Frame Force bit This bit will force the the CRC value of the transmitted data frame to 0 whenever there is a buffer overrun or underrun condition. This can be used to force a corrupted CRC as the data is not guaranteed to be reliable. The receiver will treat the data as invalid and can handle this as needed. Note: DO NOT use FORCE_ERR if using the SW CRC mode (FSI Transmit). 0h (R/W) = The CRC will not be forced to 0. 1h (R/W) = The CRC will be forced to 0 in a buffer overrun or underrun condition. Reset type: SYSRSn
4-0	RESERVED	R	0h	Reserved

36.6.2.5 TX_FRAME_CTRL Register (Offset = Ch) [Reset = 0000h]

TX_FRAME_CTRL is shown in Figure 36-18 and described in Table 36-23.

Return to the Summary Table.

Transmit frame control register

Figure 36-18. TX_FRAME_CTRL Register

15	14	13	12	11	10	9	8
START	RESERVED						
R/W-0h	R-0h						
7	6	5	4	3	2	1	0
N_WORDS				FRAME_TYPE			
R/W-0h				R/W-0h			

Table 36-23. TX_FRAME_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	START	R/W	0h	Start Transmission bit This bit will cause the FSI to start transmitting the next frame. 0h (R/W) = Writing a 0 to this bit will have no effect. 1h (R/W) = Start the next transmission. This bit will be cleared by hardware. Reset type: SYSRSn
14-8	RESERVED	R	0h	Reserved
7-4	N_WORDS	R/W	0h	Number of Words to be Transmitted This field defines the number of words which will be transmitted in a DATA_N_WORD frame. This is a user-defined field that must match the corresponding field in the receiver. Set this bitfield to be one less than the number of words to be transmitted. 0h (R/W) = 1 data word frame (16-bit data). 1h (R/W) = 2 data word frame (32-bit data). .. Fh (R/W) = 16 data word frame (256-bit data). Reset type: SYSRSn
3-0	FRAME_TYPE	R/W	0h	Transmit Frame Type This field determines the type of frame that will be transmitted next. 0000b (R/W) = Ping Frame. This frame can be sent either by software or automatically by hardware. 0100b (R/W) = DATA_1_WORD Frame. One word data frame (16-bit data). 0101b (R/W) = DATA_2_WORD Frame. Two word data frame (32-bit data). 0110b (R/W) = DATA_4_WORD Frame. Four word data frame (64-bit data). 0111b (R/W) = DATA_6_WORD Frame. Six word data frame (96-bit data). 0011b (R/W) = DATA_N_WORD Frame. The N_WORDS field will determine the number of words (1 to 16) to be sent. Both the transmitter and receiver must have the same value programmed. 1111b (R/W) = Error Frame. This frame can be used during error conditions or any condition where the transmitter wants to notify the receiver of a high priorty status. However, the user software is at liberty to use this for any purpose. 0001b, 0010b, and 1000b through 1110b are Reserved and should not be used. Reset type: SYSRSn

36.6.2.6 TX_FRAME_TAG_UDATA Register (Offset = Eh) [Reset = 0000h]

TX_FRAME_TAG_UDATA is shown in Figure 36-19 and described in Table 36-24.

Return to the Summary Table.

Transmit frame tag and user data register

Figure 36-19. TX_FRAME_TAG_UDATA Register

15	14	13	12	11	10	9	8
USER_DATA							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				FRAME_TAG			
R-0h				R/W-0h			

Table 36-24. TX_FRAME_TAG_UDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	USER_DATA	R/W	0h	User Data bits This is a user-defined value that will be loaded into the the user data phase of the frame. This 8-bit value can be used by the receiver for any application need. This value will not impact any hardware behavior. Reset type: SYSRSn
7-4	RESERVED	R	0h	Reserved
3-0	FRAME_TAG	R/W	0h	This will be used only for software initiated transmissions. Frame tag bits This is a user-defined value that will be loaded into the frame tag phase of the next transmission. The receiver may use the frame tag for any application need. This value will not impact any hardware behavior For external triggers do not use this register. Use the TX_PING_TAG register instead. Reset type: SYSRSn

36.6.2.7 TX_BUF_PTR_LOAD Register (Offset = 10h) [Reset = 0000h]

TX_BUF_PTR_LOAD is shown in Figure 36-20 and described in Table 36-25.

Return to the Summary Table.

Transmit buffer pointer control load register

Figure 36-20. TX_BUF_PTR_LOAD Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				BUF_PTR_LOAD			
R-0h				R/W-0h			

Table 36-25. TX_BUF_PTR_LOAD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-0	BUF_PTR_LOAD	R/W	0h	Buffer Pointer Load bits These bits are used to force the transmit buffer pointer to a desired index within the transmit buffer. The next transmission will begin picking data from this index and increment appropriately. This value will be reflected in TX_BUF_PTR_STS only after a minimum 3 SYSCLK cycles + 3 TXCLK cycles. This value should not be written while there is an active transmission as it may corrupt the ongoing frame or other undefined behavior. Reset type: SYSRSn

36.6.2.8 TX_BUF_PTR_STS Register (Offset = 12h) [Reset = 0000h]

TX_BUF_PTR_STS is shown in Figure 36-21 and described in Table 36-26.

Return to the Summary Table.

Transmit buffer pointer control status register

Figure 36-21. TX_BUF_PTR_STS Register

15	14	13	12	11	10	9	8
RESERVED			CURR_WORD_CNT				
R-0h			R-0h				
7	6	5	4	3	2	1	0
RESERVED				CURR_BUF_PTR			
R-0h				R-0h			

Table 36-26. TX_BUF_PTR_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12-8	CURR_WORD_CNT	R	0h	Words Remaining in the transmit buffer This value indicates the number of words present in the data buffer which have not yet been transmitted. This value is only valid when there is no active transmission. Note: This value will not be valid if there is a buffer overrun or underrun condition. Reset type: SYSRSn
7-4	RESERVED	R	0h	Reserved
3-0	CURR_BUF_PTR	R	0h	Current Buffer Pointer Index This bitfield will show the current index of the buffer pointer. This value is only valid when there is no active transmission. Reset type: SYSRSn

36.6.2.9 TX_PING_CTRL Register (Offset = 14h) [Reset = 0000h]

TX_PING_CTRL is shown in Figure 36-22 and described in Table 36-27.

Return to the Summary Table.

Transmit ping control register

Figure 36-22. TX_PING_CTRL Register

15	14	13	12	11	10	9	8
RESERVED						EXT_TRIG_SEL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
EXT_TRIG_SEL					EXT_TRIG_EN	TIMER_EN	CNT_RST
R/W-0h					R/W-0h	R/W-0h	R/W-0h

Table 36-27. TX_PING_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-3	EXT_TRIG_SEL	R/W	0h	External Trigger Select bits This bitfield will select one of the 128 external trigger inputs to as the source to generate a ping frame. A ping frame will only be generated if the EXT_TRIG_EN bit is set. 0h (R/W) = Trigger 1 will be used to generate a ping frame. 1h (R/W) = Trigger 2 will be used to generate a ping frame. .. 7Fh (R/W) = Trigger 128 will be used to generate a ping frame. Reset type: SYSRSn
2	EXT_TRIG_EN	R/W	0h	External Trigger Enable bit This bit will allow the external trigger logic to generate a ping frame. 0h (R/W) = External triggers will not be used to generate ping frames. 1h (R/W) = The selected external trigger (selected by EXT_TRIG_SEL bits) will be able to generate a ping frame. The ping timer will be ignored if this bit is set. Reset type: SYSRSn
1	TIMER_EN	R/W	0h	Ping Timer Enable bit This bit will enable the ping timer for generating periodic ping frames. 0h (R/W) = The ping timer is disabled and will not generate ping frames. 1h (R/W) = The ping timer is enabled and can be used to generate ping frames.Once the timer count reaches the value set by the TX_PING_TO_REF register, it will initiate a ping frame transmission. Note: If the ping timer is used, EXT_TRIG_EN should not be set as it will override this function. Reset type: SYSRSn
0	CNT_RST	R/W	0h	Ping Counter Reset bit Writing a 1 to this bit will reset the ping counter to 0. The counter will stay in reset as long as this bit is set to 1. This bit needs to be cleared to 0 to use the counter. 0h (R/W) = Clear the CNT_RST. 1h (R/W) = The ping counter will be reset to 0. Reset type: SYSRSn

36.6.2.10 TX_PING_TAG Register (Offset = 16h) [Reset = 0000h]

TX_PING_TAG is shown in Figure 36-23 and described in Table 36-28.

Return to the Summary Table.

Transmit ping tag register

Figure 36-23. TX_PING_TAG Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				TAG			
R-0h				R/W-0h			

Table 36-28. TX_PING_TAG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-0	TAG	R/W	0h	Ping Frame Tag This field contains a 4-bit tag which will be sent in any ping frame that is initiated by an external trigger or the ping timer. This field is user-defined and can be set based on the application requirement. If a ping frame is generated manually, the transmitted tag will be from TX_FRAME_TAG_UDATA.FRAME_TAG, not this value. Reset type: SYSRSn

36.6.2.11 TX_PING_TO_REF Register (Offset = 18h) [Reset = 00000000h]

TX_PING_TO_REF is shown in Figure 36-24 and described in Table 36-29.

Return to the Summary Table.

Transmit ping timeout counter reference

Figure 36-24. TX_PING_TO_REF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO_REF																															
R/W-0h																															

Table 36-29. TX_PING_TO_REF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TO_REF	R/W	0h	Ping Timer Reference Value. This is the 32-bit reference value for the ping timer. The timer will increment the counter starting from 0. When the reference value is reached, it will generate a timeout event, triggering a ping frame transmission. The counter will then reset to 0 and continue counting. Reset type: SYSRSn

36.6.2.12 TX_PING_TO_CNT Register (Offset = 1Ch) [Reset = 00000000h]

TX_PING_TO_CNT is shown in Figure 36-25 and described in Table 36-30.

Return to the Summary Table.

Transmit ping timeout current count

Figure 36-25. TX_PING_TO_CNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO_CNT																															
R-0h																															

Table 36-30. TX_PING_TO_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TO_CNT	R	0h	Ping Timer Counter Value This register contains the current value of the ping timer counter. After reset, this counter will increment until it reaches the reference value (TX_PING_TO_REF), at which point it generates a ping frame transmission. After this point, the counter will reset to 0 and continue counting. This is a free-running counter Reset type: SYSRSn

36.6.2.13 TX_INT_CTRL Register (Offset = 20h) [Reset = 0000h]

TX_INT_CTRL is shown in Figure 36-26 and described in Table 36-31.

Return to the Summary Table.

Transmit interrupt event control register

Figure 36-26. TX_INT_CTRL Register

15	14	13	12	11	10	9	8
RESERVED				INT2_EN_PING_TO	INT2_EN_BUF_OVERRUN	INT2_EN_BUF_UNDERRUN	INT2_EN_FRAME_DONE
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED				INT1_EN_PING_TO	INT1_EN_BUF_OVERRUN	INT1_EN_BUF_UNDERRUN	INT1_EN_FRAME_DONE
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 36-31. TX_INT_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11	INT2_EN_PING_TO	R/W	0h	Enable PING Timer Interrupt to INT2 This bit allows the event to generate an interrupt on the INT2 line. 0h (R/W) = This event will not trigger an interrupt on TX_INT2. 1h (R/W) = The ping timer event will trigger an interrupt on TX_INT2. Reset type: SYSRSn
10	INT2_EN_BUF_OVERRUN	R/W	0h	Enable Buffer Overrun Interrupt to INT2 This bit allows the event to generate an interrupt on the INT2 line. 0h (R/W) = This event will not trigger an interrupt on TX_INT2. 1h (R/W) = A Buffer Overrun condition will trigger an interrupt on TX_INT2. Reset type: SYSRSn
9	INT2_EN_BUF_UNDERRUN	R/W	0h	Enable Buffer Underrun Interrupt to INT2 This bit allows the event to generate an interrupt on the INT2 line. 0h (R/W) = This event will not trigger an interrupt on TX_INT2. 1h (R/W) = A Buffer Underrun condition will trigger an interrupt on TX_INT2. Reset type: SYSRSn
8	INT2_EN_FRAME_DONE	R/W	0h	Enable Frame Done interrupt to INT2 This bit allows the event to generate an interrupt on the INT2 line. 0h (R/W) = This event will not trigger an interrupt on TX_INT2. 1h (R/W) = A Frame Done event will trigger an interrupt on TX_INT2. Reset type: SYSRSn
7-4	RESERVED	R	0h	Reserved
3	INT1_EN_PING_TO	R/W	0h	Enable Ping Timer Interrupt to INT1 This bit allows the event to generate an interrupt on the INT1 line. 0h (R/W) = This event will not trigger an interrupt on TX_INT1. 1h (R/W) = The ping timer event will trigger an interrupt on TX_INT1. Reset type: SYSRSn
2	INT1_EN_BUF_OVERRUN	R/W	0h	Enable Buffer Overrun Interrupt to INT1 This bit allows the event to generate an interrupt on the INT1 line. 0h (R/W) = This event will not trigger an interrupt on TX_INT1. 1h (R/W) = A Buffer Overrun condition will trigger an interrupt on TX_INT1. Reset type: SYSRSn

Table 36-31. TX_INT_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	INT1_EN_BUF_UNDERRUN	R/W	0h	Enable Buffer Underrun Interrupt to INT1 This bit allows the event to generate an interrupt on the INT1 line. 0h (R/W) = This event will not trigger an interrupt on TX_INT1. 1h (R/W) = A Buffer Underrun condition will trigger an interrupt on TX_INT1. Reset type: SYSRSn
0	INT1_EN_FRAME_DONE	R/W	0h	Enable Frame Done interrupt to INT1 This bit allows the event to generate an interrupt on the INT1 line. 0h (R/W) = This event will not trigger an interrupt on TX_INT1. 1h (R/W) = A Frame Done event will trigger an interrupt on TX_INT1. Reset type: SYSRSn

36.6.2.14 TX_DMA_CTRL Register (Offset = 22h) [Reset = 0000h]

TX_DMA_CTRL is shown in Figure 36-27 and described in Table 36-32.

Return to the Summary Table.

Transmit DMA event control register

Figure 36-27. TX_DMA_CTRL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							DMA_EVT_EN
R-0h							R/W-0h

Table 36-32. TX_DMA_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R	0h	Reserved
0	DMA_EVT_EN	R/W	0h	DMA Event Enable bit This bit will enable the DMA event to be generated upon the completion of a transmit frame. 0h (R/W) = A DMA event will not be generated. 1h (R/W) = A DMA event will be generated upon the completion of a transmitted frame. Note: The DMA event will only be generated for data frames. Reset type: SYSRSn

36.6.2.15 TX_LOCK_CTRL Register (Offset = 24h) [Reset = 0000h]

TX_LOCK_CTRL is shown in Figure 36-28 and described in Table 36-33.

Return to the Summary Table.

Transmit lock control register

Figure 36-28. TX_LOCK_CTRL Register

15	14	13	12	11	10	9	8
KEY							
W-0h							
7	6	5	4	3	2	1	0
RESERVED							LOCK
R-0h							R/W-0h

Table 36-33. TX_LOCK_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	KEY	W	0h	Write Key In order to write to this register, 0xA5 must be written to this field at the same time. Otherwise, writes are ignored. The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
7-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	Control Register Lock Enable bit This bit locks the contents of all the transmit control registers that support a lock protection. Once locked, further writes will not take effect until a SYSRS has reset this register. Once set, further writes to this bit will be ignored. 0h (R/W) = Transmit control registers can be modified and are not locked. 1h (R/W) = Transmit control registers are locked and cannot be modified until this bit is cleared by SYSRS. Any further writes to this bit are ignored. Note: The KEY field must contatin 0xA5 for any write to this bit to take effect. Reset type: SYSRSn

36.6.2.16 TX_EVT_STS Register (Offset = 28h) [Reset = 0000h]

TX_EVT_STS is shown in Figure 36-29 and described in Table 36-34.

Return to the Summary Table.

Transmit event and error status flag register

Figure 36-29. TX_EVT_STS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				PING_TRIGGERED	BUF_OVERRUN	BUF_UNDERRUN	FRAME_DONE
R-0h				R-0h	R-0h	R-0h	R-0h

Table 36-34. TX_EVT_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3	PING_TRIGGERED	R	0h	Ping Frame Triggered Flag Bit This bit indicates that a ping frame has been triggered. This bit is set by hardware when either the ping timer or an external trigger event have occured. Software can also force this bit to get set by writing to the TX_EVT_FRC register. 0h (R) = A ping frame has not been triggered. 1h (R) = A ping frame has been triggered by either the ping timer or external trigger. To clear this bit, write to the corresponding bit in the TX_EVT_CLR register. Reset type: SYSRSn
2	BUF_OVERRUN	R	0h	Buffer Overrun Flag Bit This bit inditcates that buffer overrun has occured.Software can also force this bit to get set by writing to the TX_EVT_FRC register. 0h (R) = Buffer Overrun has not occured. 1h (R) = Buffer Overrun has occured. To clear this bit, write to the corresponding bit in the TX_EVT_CLR register. Reset type: SYSRSn
1	BUF_UNDERRUN	R	0h	Buffer Underrun Flag Bit This bit inditcates that buffer underrun has occured.Software can also force this bit to get set by writing to the TX_EVT_FRC register. 0h (R) = Buffer Underrun has not occured. 1h (R) = Buffer Underrun has occured. To clear this bit, write to the corresponding bit in the TX_EVT_CLR register. Reset type: SYSRSn
0	FRAME_DONE	R	0h	Frame Done Flag Bit This bit inditcates a Frame Done condition. This bit is set by hardware when a frame transmission has been completed. Software can also force this bit to get set by writing to the TX_EVT_FRC register. 0h (R) = Frame Done condition has not occured. 1h (R) = Frame Done condition has occured. To clear this bit, write to the corresponding bit in the TX_EVT_CLR register. Reset type: SYSRSn

36.6.2.17 TX_EVT_CLR Register (Offset = 2Ch) [Reset = 0000h]

TX_EVT_CLR is shown in Figure 36-30 and described in Table 36-35.

Return to the Summary Table.

Transmit event and error clear register

Figure 36-30. TX_EVT_CLR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				PING_TRIGGERED	BUF_OVERRUN	BUF_UNDERRUN	FRAME_DONE
R-0h				W-0h	W-0h	W-0h	W-0h

Table 36-35. TX_EVT_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3	PING_TRIGGERED	W	0h	Ping Frame Triggered Flag Clear bit This bit clears the corresponding bit in the TX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the TX_EVT_STS register to 0. Note: This bit may not always be cleared when writing to the corresponding TX_EVT_CLR bit. If PING_TIMEOUT MODE is configured to be 0, a hardware ping timeout may occur when another frame is actively being transmitted. In this case, if this bit still shows as 1 after the clear bit is written then the ping frame has been triggered but not serviced. This bit does not indicate that the ping frame has been completely sent, only that it has been triggered by the timeout event. Reset type: SYSRSn
2	BUF_OVERRUN	W	0h	Buffer Overrun Flag Clear bit This bit clears the corresponding bit in the TX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the TX_EVT_STS register to 0. Reset type: SYSRSn
1	BUF_UNDERRUN	W	0h	Buffer Underrun Flag Clear bit This bit clears the corresponding bit in the TX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the TX_EVT_STS register to 0. Reset type: SYSRSn
0	FRAME_DONE	W	0h	Frame Done Flag Clear bit This bit clears the corresponding bit in the TX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the TX_EVT_STS register to 0. Reset type: SYSRSn

36.6.2.18 TX_EVT_FRC Register (Offset = 2Eh) [Reset = 0000h]

TX_EVT_FRC is shown in Figure 36-31 and described in Table 36-36.

Return to the Summary Table.

Transmit event and error flag force register

Figure 36-31. TX_EVT_FRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				PING_TRIGGERED	BUF_OVERRUN	BUF_UNDERRUN	FRAME_DONE
R-0h				W-0h	W-0h	W-0h	W-0h

Table 36-36. TX_EVT_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3	PING_TRIGGERED	W	0h	Ping Frame Triggered Flag Force bit This bit will cause the corresponding bit in the TX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding flag bit in the TX_EVT_STS Register. Reset type: SYSRSn
2	BUF_OVERRUN	W	0h	Buffer Overrun Flag Force bit This bit will cause the corresponding bit in the TX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (R/W) = Writing a 0 to this bit will have no effect. 1h (R/W) = Force the corresponding flag bit in the TX_EVT_STS Register. Reset type: SYSRSn
1	BUF_UNDERRUN	W	0h	Buffer Underrun Flag Force bit This bit will cause the corresponding bit in the TX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding flag bit in the TX_EVT_STS Register. Reset type: SYSRSn
0	FRAME_DONE	W	0h	Frame Done Flag Force bit This bit will cause the corresponding bit in the TX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding flag bit in the TX_EVT_STS Register. Reset type: SYSRSn

36.6.2.19 TX_USER_CRC Register (Offset = 30h) [Reset = 0000h]

TX_USER_CRC is shown in Figure 36-32 and described in Table 36-37.

Return to the Summary Table.

Transmit user-defined CRC register

Figure 36-32. TX_USER_CRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
USER_CRC							
R/W-0h							

Table 36-37. TX_USER_CRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	USER_CRC	R/W	0h	User-defined CRC This register contains the 8-bit CRC value to be transmitted in the next frame if the transmission is set for user-defined CRC option (TX_OPER_CTRL_LO.SW_CRC = 1). This register is ignored if the hardware CRC generation is enabled. Reset type: SYSRSn

36.6.2.20 TX_ECC_DATA Register (Offset = 40h) [Reset = 00000000h]

TX_ECC_DATA is shown in Figure 36-33 and described in Table 36-38.

Return to the Summary Table.

Transmit ECC data register

Figure 36-33. TX_ECC_DATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_HIGH																DATA_LOW															
R/W-0h																R/W-0h															

Table 36-38. TX_ECC_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DATA_HIGH	R/W	0h	Upper 16 bits of ECC Data Writing to this bitfield will cause the ECC logic to compute the ECC(SEC-DED) the entire 32-bit register and update TX_ECC_VAL register with the results. Software should write to these 16 bits of the register in a 32-bit write when needing to compute ECC for 32-bits for the full TX_ECC_DATA register. Reset type: SYSRSn
15-0	DATA_LOW	R/W	0h	Lower 16 bits of ECC Data Writing to this bitfield will cause the ECC logic to compute the ECC(SEC-DED) for these 16 bits and update the TX_ECC_VAL register with the results. Software should write to these register bits as a 16-bit write when needing to compute ECC for 16-bits. Reset type: SYSRSn

36.6.2.21 TX_ECC_VAL Register (Offset = 44h) [Reset = 000Ch]

TX_ECC_VAL is shown in Figure 36-34 and described in Table 36-39.

Return to the Summary Table.

Transmit ECC value register

Figure 36-34. TX_ECC_VAL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	ECC_VAL						
R-0h	R-Ch						

Table 36-39. TX_ECC_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6-0	ECC_VAL	R	Ch	Computed ECC Value This field contains the ECC value computed using SEC-DED either for 16-bit or 32-bit data in the TX_ECC_DATA register. Reset type: SYSRSn

36.6.2.22 TX_DLYLINE_CTRL Register (Offset = 48h) [Reset = 0000h]

TX_DLYLINE_CTRL is shown in Figure 36-35 and described in Table 36-40.

Return to the Summary Table.

Transmit delay Line control register

Figure 36-35. TX_DLYLINE_CTRL Register

15	14	13	12	11	10	9	8
RESERVED	TXD1_DLY					TXD0_DLY	
R-0h	R/W-0h					R/W-0h	
7	6	5	4	3	2	1	0
TXD0_DLY			TXCLK_DLY				
R/W-0h			R/W-0h				

Table 36-40. TX_DLYLINE_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14-10	TXD1_DLY	R/W	0h	Delay Line Tap Select for TXD1 This bitfield selects the number of delay elements inserted into the TXD1 path from the pin boundary to the receiver core. 0h (R/W) Zero delay elements are included in the TXD1 path. TXD1 is taken directly from the pin. 1h (R/W) One delay element is included in the TXD1 path. 2h (R/W) Two delay elements are included in the TXD1 path. ... 1Fh (R/W) 31 delay elements are included in the TXD1 path, the maximum. Reset type: SYSRSn
9-5	TXD0_DLY	R/W	0h	Delay Line Tap Select for TXD0 This bitfield selects the number of delay elements inserted into the TXD0 path from the pin boundary to the receiver core. 0h (R/W) Zero delay elements are included in the TXD0 path. TXD0 is taken directly from the pin. 1h (R/W) One delay element is included in the TXD0 path. 2h (R/W) Two delay elements are included in the TXD0 path. ... 1Fh (R/W) 31 delay elements are included in the TXD0 path, the maximum. Reset type: SYSRSn
4-0	TXCLK_DLY	R/W	0h	Delay Line Tap Select for TXCLK This bitfield selects the number of delay elements inserted into the TXCLK path from the pin boundary to the receiver core. 0h (R/W) Zero delay elements are included in the TXCLK path. TXCLK is taken directly from the pin. 1h (R/W) One delay element is included in the TXCLK path. 2h (R/W) Two delay elements are included in the TXCLK path. ... 1Fh (R/W) 31 delay elements are included in the TXCLK path, the maximum. Reset type: SYSRSn

36.6.2.23 TX_BUF_BASE_y Register (Offset = 80h + formula) [Reset = 0000h]

TX_BUF_BASE_y is shown in Figure 36-36 and described in Table 36-41.

Return to the Summary Table.

Base address for transmit buffer

Offset = 80h + (y * 2h); where y = 0h to Fh

Figure 36-36. TX_BUF_BASE_y Register

15	14	13	12	11	10	9	8
BASE_ADDRESS							
R/W-0h							
7	6	5	4	3	2	1	0
BASE_ADDRESS							
R/W-0h							

Table 36-41. TX_BUF_BASE_y Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	BASE_ADDRESS	R/W	0h	Transmit Data Buffer Base Address This is the base address of the 16-word data buffer used by the transmitter. Reset type: SYSRSn

36.6.3 FSI_RX_REGS Registers

Table 36-42 lists the memory-mapped registers for the FSI_RX_REGS registers. All register offset addresses not listed in Table 36-42 should be considered as reserved locations and the register contents should not be modified.

Table 36-42. FSI_RX_REGS Registers

Offset	Acronym	Register Name	Protection
0h	RX_MAIN_CTRL	Receive main control register	
8h	RX_OPER_CTRL	Receive operation control register	and LOCK
Ch	RX_FRAME_INFO	Receive frame control register	
Eh	RX_FRAME_TAG_UDATA	Receive frame tag and user data register	
10h	RX_DMA_CTRL	Receive DMA event control register	and LOCK
14h	RX_EVT_STS	Receive event and error status flag register	
16h	RX_CRC_INFO	Receive CRC info of received and computed CRC	
18h	RX_EVT_CLR	Receive event and error clear register	
1Ah	RX_EVT_FRC	Receive event and error flag force register	
1Ch	RX_BUF_PTR_LOAD	Receive buffer pointer load register	
1Eh	RX_BUF_PTR_STS	Receive buffer pointer status register	
20h	RX_FRAME_WD_CTRL	Receive frame watchdog control register	and LOCK
24h	RX_FRAME_WD_REF	Receive frame watchdog counter reference	and LOCK
28h	RX_FRAME_WD_CNT	Receive frame watchdog current count	
2Ch	RX_PING_WD_CTRL	Receive ping watchdog control register	and LOCK
2Eh	RX_PING_TAG	Receive ping tag register	
30h	RX_PING_WD_REF	Receive ping watchdog counter reference	and LOCK
34h	RX_PING_WD_CNT	Receive pingwatchdog current count	
38h	RX_INT1_CTRL	Receive interrupt control register for RX_INT1	and LOCK
3Ah	RX_INT2_CTRL	Receive interrupt control register for RX_INT2	and LOCK
3Ch	RX_LOCK_CTRL	Receive lock control register	and LOCK
40h	RX_ECC_DATA	Receive ECC data register	
44h	RX_ECC_VAL	Receive ECC value register	
48h	RX_ECC_SEC_DATA	Receive ECC corrected data register	
4Ch	RX_ECC_LOG	Receive ECC log and status register	
50h	RX_FRAME_TAG_CMP	Receive frame tag compare register	and LOCK
52h	RX_PING_TAG_CMP	Receive ping tag compare register	and LOCK
58h	RX_TRIG_CTRL_0	Receive Trigger Control register 0	and LOCK
5Ch	RX_TRIG_WIDTH_0	Receive Trigger Wdith register 0	and LOCK
60h	RX_DLYLINE_CTRL	Receive delay line control register	and LOCK
64h	RX_TRIG_CTRL_1	Receive Trigger Control register 1	and LOCK
68h	RX_TRIG_CTRL_2	Receive Trigger Control register 2	and LOCK
6Ch	RX_TRIG_CTRL_3	Receive Trigger Control register 3	and LOCK
70h	RX_VIS_1	Receive debug visibility register 1	
74h	RX_UDATA_FILTER	Receive User Data Filter Control register	and LOCK
80h + formula	RX_BUF_BASE_y	Base address for receive data buffer	

Complex bit access types are encoded to fit into small table cells. Table 36-43 shows the codes that are used for access types in this section.

Table 36-43. FSI_RX_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

36.6.3.1 RX_MAIN_CTRL Register (Offset = 0h) [Reset = 0000h]

RX_MAIN_CTRL is shown in Figure 36-37 and described in Table 36-44.

Return to the Summary Table.

Receive main control register

Figure 36-37. RX_MAIN_CTRL Register

15	14	13	12	11	10	9	8
KEY							
W-0h							
7	6	5	4	3	2	1	0
RESERVED			DATA_FILTER_EN	INPUT_ISOLATE	SPI_PAIRING	INT_LOOPBACK	CORE_RST
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 36-44. RX_MAIN_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	KEY	W	0h	Write Key. In order to write to this register, 0xA5 must be written to this field at the same time. Otherwise, writes are ignored. The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
7-5	RESERVED	R	0h	Reserved
4	DATA_FILTER_EN	R/W	0h	Data Filter Enable Bit. 0h (R/W) = Data filtering is disabled. 1h (R/W) = Data filtering is enabled. Reset type: SYSRSn
3	INPUT_ISOLATE	R/W	0h	When set to 1, the FSI RX inputs (RXCLK, RXD0 and RXD1) will be isolated from what is driven from the device pins and will be held at inactive level of '1'.This isolation facilitates the user to switch the RX inputs to a different set of device pins and hence any potential glitch that could occur during the process of switching will not affect the RX module itself. Reset type: SYSRSn
2	SPI_PAIRING	R/W	0h	Clock Pairing for SPI-like Behavior Enable bit This bit enables the internal clock pairing with the FSI TX module. This feature internally connects the TXCLK to RXCLK allowing the FSI TX module, acting as a SPI controller, to clock data into the receiver and out of the transmitter like a standard SPI module. This configuration is valid when the Module is in SPI mode only (RX_OPER_CTRL.SPI_MODE = 1) 0h (R/W) = SPI clock pairing is not enabled. 1h (R/W) = SPI clock pairing is enabled. The RXCLK will be internally connected to the TXCLK of the corresponding FSI module. Note: The KEY field must contatin 0xA5 for any write to this bit to take effect. Reset type: SYSRSn

Table 36-44. RX_MAIN_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	INT_LOOPBACK	R/W	0h	Internal Loopback Enable bit This bit enables the internal loopback functionality of the FSI receiver. By enabling this bit, a mux will select the signals coming directly from the corresponding FSI transmitter module rather than from the pins. 0h (R/W) = Internal loopback is disabled. The FSI RX module will receive signals coming from the pins. 1h (R/W) = Internal loopback is enabled. The FSI RX module will receive signals from the directly from FSI TX module rather than the pins. Note: The KEY field must contatin 0xA5 for any write to this bit to take effect. Reset type: SYSRSn
0	CORE_RST	R/W	0h	Receiver Main Core Reset bit This bit controls the receiver main core reset. In order to receive any frame, this bit must be cleared. Note: For reset to take affect, the FSI RX module must be held in reset for at least 4 SYSCLK cycles. 0h (R/W) = Receiver core is not in reset and can receive frames. 1h (R/W) = Receiver core is held in reset. Note: The KEY field must contatin 0xA5 for any write to this bit to take effect. Reset type: SYSRSn

36.6.3.2 RX_OPER_CTRL Register (Offset = 8h) [Reset = 0000h]

RX_OPER_CTRL is shown in Figure 36-38 and described in Table 36-45.

Return to the Summary Table.

Receive operation control register

Figure 36-38. RX_OPER_CTRL Register

15	14	13	12	11	10	9	8
RESERVED							PING_WD_RST_MODE
R-0h							R/W-0h
7	6	5	4	3	2	1	0
ECC_SEL	N_WORDS				SPI_MODE	DATA_WIDTH	
R/W-0h	R/W-0h				R/W-0h	R/W-0h	

Table 36-45. RX_OPER_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8	PING_WD_RST_MODE	R/W	0h	Ping Watchdog Timeout Mode Select bit This bit selects the mode by which the ping watchdog counter is reset. The watchdog counter can be reset and restarted only by ping frames or by any received frame. 0h (R/W) = The ping watchdog counter will reset and restart only by ping frames. 1h (R/W) = The ping watchdog counter will reset and restart by any received frame. Reset type: SYSRSn
7	ECC_SEL	R/W	0h	ECC Data Width Select bit This bit selects between whether the ECC computation is done on 16-bit or 32-bit words. 0h (R/W) = 32-bit ECC is used. 1h (R/W) = 16-bit ECC is used. Reset type: SYSRSn
6-3	N_WORDS	R/W	0h	Number of Words to Receive This field defines the number of words which will be received in a DATA_N_WORD frame. This is a user-defined field that must match the corresponding field in the transmitter. Set this bitfield to be one less than the number of words to be received. This value is only applicable when the frame type received is DATA_N_WORD. 0h (R/W) = 1 data word frame (16-bit data). 1h (R/W) = 2 data word frame (32-bit data). .. Fh (R/W) = 16 data word frame (256-bit data). Reset type: SYSRSn
2	SPI_MODE	R/W	0h	SPI Mode Enable bit This bit enables and disables the SPI compatibility mode of the FSI RX. The received data must be formatted as an FSI frame in order for the data to properly be received. SPI compatibility mode will allow FSI RX to receive data that is sent using SPI signal format. Refer to the applicable section in the FSI TRM chapter for more information. 0h (R/W) = FSI is in normal mode of operation. 1h (R/W) = FSI is operating in SPI compatibility mode. Reset type: SYSRSn

Table 36-45. RX_OPER_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	DATA_WIDTH	R/W	0h	Receive Data Width Select bit These bits decide the number of data lines used for receiving data. 0h (R/W) = Data will be received on one data line, RXD0. 1h (R/W) = Data will be received on two data lines, RXD0 and RXD1. 2h, 3h (R/W) = Reserved Reset type: SYSRSn

36.6.3.3 RX_FRAME_INFO Register (Offset = Ch) [Reset = 0000h]

RX_FRAME_INFO is shown in Figure 36-39 and described in Table 36-46.

Return to the Summary Table.

Receive frame control register

Figure 36-39. RX_FRAME_INFO Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				FRAME_TYPE			
R-0h				R-0h			

Table 36-46. RX_FRAME_INFO Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-0	FRAME_TYPE	R	0h	Received Frame Type This field indicates the type of non-ping frame that was successfully received last. Note: Ping frame reception does not update this field, we want to retain the last successful non-ping frame FRAME_TYPE and PING_FRAME_RCVD flag already conveys PING info to the user. 0100b (R/W) = A DATA_1_WORD frame was received (16-bit data). 0101b (R/W) = A DATA_2_WORD frame was received (32-bit data). 0110b (R/W) = A DATA_4_WORD frame was received (64-bit data). 0111b (R/W) = A DATA_6_WORD frame was received (96-bit data). 0011b (R/W) = A DATA_N_WORD frame was received. The N_WORD field will determine the number of words (1 to 16) to be sent. The number of words received must equal the value programmed in RX_OPER_CTRL.N_WORDS. 1111b (R/W) = An error frame was received. This frame can be used during error conditions or any condition where the transmitter wants to signal the receiver for attention. However, the user software is at liberty to use this for any purpose. 0001b, 0010b, and 1000b through 1110b are Reserved and should not be used. Reset type: SYSRSn

36.6.3.4 RX_FRAME_TAG_UDATA Register (Offset = Eh) [Reset = 0000h]

RX_FRAME_TAG_UDATA is shown in Figure 36-40 and described in Table 36-47.

Return to the Summary Table.

Receive frame tag and user data register

Figure 36-40. RX_FRAME_TAG_UDATA Register

15	14	13	12	11	10	9	8
USER_DATA							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			FRAME_TAG				RESERVED
R-0h			R-0h				R-0h

Table 36-47. RX_FRAME_TAG_UDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	USER_DATA	R	0h	Received User Data This field contains the 8-bit user data field of the last successfully received frame. Reset type: SYSRSn
7-5	RESERVED	R	0h	Reserved
4-1	FRAME_TAG	R	0h	Received Frame Tag This field contains the 4-bit frame tag from the last successfully received frame. This is intentionally shifted into bits 4:1 so that the register can be used as a 32-bit address index based on the received tag. Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

36.6.3.5 RX_DMA_CTRL Register (Offset = 10h) [Reset = 0000h]

RX_DMA_CTRL is shown in Figure 36-41 and described in Table 36-48.

Return to the Summary Table.

Receive DMA event control register

Figure 36-41. RX_DMA_CTRL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							DMA_EVT_EN
R-0h							R/W-0h

Table 36-48. RX_DMA_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R	0h	Reserved
0	DMA_EVT_EN	R/W	0h	DMA Event Enable bit This bit will enable a DMA Event to be generated upon the completion of a frame reception. 0h (R/W) = A DMA event will not be generated. 1h (R/W) = A DMA event will be generated upon the reception of a frame. Note: The DMA event will only be generated for data frames. Reset type: SYSRSn

36.6.3.6 RX_EVT_STS Register (Offset = 14h) [Reset = 0000h]

RX_EVT_STS is shown in Figure 36-42 and described in Table 36-49.

Return to the Summary Table.

Receive event and error status flag register

Figure 36-42. RX_EVT_STS Register

15	14	13	12	11	10	9	8
RESERVED	ERROR_TAG_MATCH	DATA_TAG_MATCH	PING_TAG_MATCH	DATA_FRAME	FRAME_OVERRUN	PING_FRAME	ERR_FRAME
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
BUF_UNDERRUN	FRAME_DONE	BUF_OVERRUN	EOF_ERR	TYPE_ERR	CRC_ERR	FRAME_WD_TO	PING_WD_TO
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 36-49. RX_EVT_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	ERROR_TAG_MATCH	R	0h	Error Tag Match Flag This bit indicates that an error frame was received with a tag comparison matching the masked tag reference. Software can also force this bit to get set by writing to the RX_EVT_FRC register. 0h (R) = No tag-matched error frame received. 1h (R) = A tag-matched error frame has been received. To clear this bit, write to the corresponding bit in the RX_EVT_CLR register. Reset type: SYSRSn
13	DATA_TAG_MATCH	R	0h	Data Tag Match Flag This bit indicates that a dataframe was received with a tag comparison matching the masked tag reference. Software can also force this bit to get set by writing to the RX_EVT_FRC register. 0h (R) = No tag-matched data frame received. 1h (R) = A tag-matched data frame has been received. To clear this bit, write to the corresponding bit in the RX_EVT_CLR register. Reset type: SYSRSn
12	PING_TAG_MATCH	R	0h	Ping Tag Match Flag This bit indicates that a ping frame was received with a tag comparison matching the masked tag reference. Software can also force this bit to get set by writing to the RX_EVT_FRC register. 0h (R) = No tag-matched ping frame received. 1h (R) = A tag-matched ping frame has been received. To clear this bit, write to the corresponding bit in the RX_EVT_CLR register. Reset type: SYSRSn
11	DATA_FRAME	R	0h	Data Frame Received Flag This bit indicates that an data frame has been received. Software can also force this bit to get set by writing to the RX_EVT_FRC register. 0h (R) = No data frame has been received. 1h (R) = A data frame has been received. To clear this bit, write to the corresponding bit in the RX_EVT_CLR register. Reset type: SYSRSn

Table 36-49. RX_EVT_STS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	FRAME_OVERRUN	R	0h	Frame Overrun Flag This bit indicates that a frame overrun condition has occured. This bit gets set to 1 when a new DATA/ERROR frame is received and the corresponding DATA_FRAME_RCVD/ERROR_FRAME_RCVD flag is still set to 1. Software can also force this bit to get set by writing to the RX_EVT_FRC register. 0h (R) = Frame overrun has not ocurred. 1h (R) = Frame overrun has ocurred. To clear this bit, write to the corresponding bit in the RX_EVT_CLR register. Reset type: SYSRSn
9	PING_FRAME	R	0h	Ping Frame Received Flag This bit indicates that an ping frame has been received. Software can also force this bit to get set by writing to the RX_EVT_FRC register. 0h (R) = No ping frame has been received. 1h (R) = A ping frame has been received. To clear this bit, write to the corresponding bit in the RX_EVT_CLR register. Reset type: SYSRSn
8	ERR_FRAME	R	0h	Error Frame Received Flag This bit indicates that an error frame has been received. Software can also force this bit to get set by writing to the RX_EVT_FRC register. 0h (R) = No error frame has been received. 1h (R) = An error frame has been received. To clear this bit, write to the corresponding bit in the RX_EVT_CLR register. Reset type: SYSRSn
7	BUF_UNDERRUN	R	0h	Receive Buffer Underrun Flag This bit indicates that a buffer underrun condition has occured in the receive buffer. This will happen when software reads the buffer which is empty and has no valid data. Software can also force this bit to get set by writing to the RX_EVT_FRC register. 0h (R) = Receive Buffer Underrun has not ocurred. 1h (R) = Receive Buffer Underrun has ocurred. To clear this bit, write to the corresponding bit in the RX_EVT_CLR register. Reset type: SYSRSn
6	FRAME_DONE	R	0h	Frame Done Flag This bit indicates that a frame has been successfully received without error. Software can also force this bit to get set by writing to the RX_EVT_FRC register. 0h (R) = No frame has been successfully received. 1h (R) = A frame has been successfully received. To clear this bit, write to the corresponding bit in the RX_EVT_CLR register. Reset type: SYSRSn
5	BUF_OVERRUN	R	0h	Receive Buffer Overrun Flag This bit indicates that a buffer overrun condition has occured in the receive buffer. Software can also force this bit to get set by writing to the RX_EVT_FRC register. 0h (R) = Receive buffer overrun has not ocurred. 1h (R) = Receive buffer overrun has ocurred. To clear this bit, write to the corresponding bit in the RX_EVT_CLR register. Reset type: SYSRSn

Table 36-49. RX_EVT_STS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	EOF_ERR	R	0h	End-of-Frame Error Flag This bit indicates that an invalid end-of-frame bit pattern has been received. Software can also force this bit to get set by writing to the RX_EVT_FRC register. 0h (R) = Invalid end-of-frame has not been received. 1h (R) = Invalid end-of-frame has been received To clear this bit, write to the corresponding bit in the RX_EVT_CLR register. Reset type: SYSRSn
3	TYPE_ERR	R	0h	Frame Type Error Flag This bit inditcates that an invalid frame type has been received. Software can also force this bit to get set by writing to the RX_EVT_FRC register. 0h (R) = Invalid frame type has not been received. 1h (R) = Invalid frame type has been received To clear this bit, write to the corresponding bit in the RX_EVT_CLR register. Reset type: SYSRSn
2	CRC_ERR	R	0h	CRC Error Flag This bit indicates that a CRC error has occured. A CRC error will be generated on a data frame where the received CRC and the computed CRC do not match. Software can also force this bit to get set by writing to the RX_EVT_FRC register. 0h (R) = CRC error has not occured. 1h (R) = CRC error has occured. To clear this bit, write to the corresponding bit in the RX_EVT_CLR register. Reset type: SYSRSn
1	FRAME_WD_TO	R	0h	Frame Watchdog Timeout Flag This bit indicates that the frame watchdog timer has timed out. Software can also force this bit to get set by writing to the RX_EVT_FRC register. 0h (R) = Frame watchdog timeout has not occured. 1h (R) = Frame watchdog timeout has occured. To clear this bit, write to the corresponding bit in the RX_EVT_CLR register. Reset type: SYSRSn
0	PING_WD_TO	R	0h	Ping Watchdog Timeout Flag This bit indicates that the ping watchdog timer has timed out. Software can also force this bit to get set by writing to the RX_EVT_FRC register. 0h (R) = Ping watchdog timeout has not occured. 1h (R) = Ping watchdog timeout has occured. To clear this bit, write to the corresponding bit in the RX_EVT_CLR register. Reset type: SYSRSn

36.6.3.7 RX_CRC_INFO Register (Offset = 16h) [Reset = 0000h]

RX_CRC_INFO is shown in Figure 36-43 and described in Table 36-50.

Return to the Summary Table.

Receive CRC info of received and computed CRC

Figure 36-43. RX_CRC_INFO Register

15	14	13	12	11	10	9	8
CALC_CRC							
R-0h							
7	6	5	4	3	2	1	0
RX_CRC							
R-0h							

Table 36-50. RX_CRC_INFO Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	CALC_CRC	R	0h	Harware Calculated CRC Value This bitfield contains the CRC value that was calculated on the last received data. The contents of this bitfield are valid only when data frames are received. Note: The contents of this bitfield are invalid for ping and error frames. Reset type: SYSRSn
7-0	RX_CRC	R	0h	Received CRC Value This bitfield contains the CRC value that was last received a frame. The contents of this bitfield are valid only when data frames are received. Note: The contents of this bitfield are invalid for ping and error frames. Reset type: SYSRSn

36.6.3.8 RX_EVT_CLR Register (Offset = 18h) [Reset = 0000h]

RX_EVT_CLR is shown in Figure 36-44 and described in Table 36-51.

Return to the Summary Table.

Receive event and error clear register

Figure 36-44. RX_EVT_CLR Register

15	14	13	12	11	10	9	8
RESERVED	ERROR_TAG_MATCH	DATA_TAG_MATCH	PING_TAG_MATCH	DATA_FRAME	FRAME_OVERRUN	PING_FRAME	ERR_FRAME
R-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
BUF_UNDERRUN	FRAME_DONE	BUF_OVERRUN	EOF_ERR	TYPE_ERR	CRC_ERR	FRAME_WD_TO	PING_WD_TO
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 36-51. RX_EVT_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	ERROR_TAG_MATCH	W	0h	Error Tag Match Glag Clear bit This bit clears the corresponding bit in the RX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the RX_EVT_STS register to 0. Reset type: SYSRSn
13	DATA_TAG_MATCH	W	0h	Data Tag Match Flag Clear bit This bit clears the corresponding bit in the RX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the RX_EVT_STS register to 0. Reset type: SYSRSn
12	PING_TAG_MATCH	W	0h	Ping Tag Match Flag Clear bit This bit clears the corresponding bit in the RX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the RX_EVT_STS register to 0. Reset type: SYSRSn
11	DATA_FRAME	W	0h	Data Frame Received Flag Clear bit This bit clears the corresponding bit in the RX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the RX_EVT_STS register to 0. Reset type: SYSRSn
10	FRAME_OVERRUN	W	0h	Frame Overrun Flag Clear bit This bit clears the corresponding bit in the RX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the RX_EVT_STS register to 0. Reset type: SYSRSn
9	PING_FRAME	W	0h	Ping Frame Received Flag Clear bit This bit clears the corresponding bit in the RX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the RX_EVT_STS register to 0. Reset type: SYSRSn

Table 36-51. RX_EVT_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	ERR_FRAME	W	0h	Error Frame Received Flag Clear bit This bit clears the corresponding bit in the RX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the RX_EVT_STS register to 0. Reset type: SYSRSn
7	BUF_UNDERRUN	W	0h	Receive Buffer Underrun Flag Clear bit This bit clears the corresponding bit in the RX_EVT_STS register. 0h (R/W) = Writing a 0 to this bit will have no effect. 1h (R/W) = Writing a 1 to this bit will clear the corresponding bit in the RX_EVT_STS register to 0. Reset type: SYSRSn
6	FRAME_DONE	W	0h	Frame Done Flag Clear bit This bit clears the corresponding bit in the RX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the RX_EVT_STS register to 0. Reset type: SYSRSn
5	BUF_OVERRUN	W	0h	Receive Buffer Overrun Flag Clear bit This bit clears the corresponding bit in the RX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the RX_EVT_STS register to 0. Reset type: SYSRSn
4	EOF_ERR	W	0h	End-of-Frame Error Flag Clear bit This bit clears the corresponding bit in the RX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the RX_EVT_STS register to 0. Reset type: SYSRSn
3	TYPE_ERR	W	0h	Frame Type Error Flag Clear bit This bit clears the corresponding bit in the RX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the RX_EVT_STS register to 0. Reset type: SYSRSn
2	CRC_ERR	W	0h	CRC Error Flag Clear bit This bit clears the corresponding bit in the RX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the RX_EVT_STS register to 0. Reset type: SYSRSn
1	FRAME_WD_TO	W	0h	Frame Watchdog Timeout Flag Clear bit This bit clears the corresponding bit in the RX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the RX_EVT_STS register to 0. Reset type: SYSRSn
0	PING_WD_TO	W	0h	Ping Watchdog Timeout Flag Clear bit This bit clears the corresponding bit in the RX_EVT_STS register. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Writing a 1 to this bit will clear the corresponding bit in the RX_EVT_STS register to 0. Reset type: SYSRSn

36.6.3.9 RX_EVT_FRC Register (Offset = 1Ah) [Reset = 0000h]

RX_EVT_FRC is shown in Figure 36-45 and described in Table 36-52.

Return to the Summary Table.

Receive event and error flag force register

Figure 36-45. RX_EVT_FRC Register

15	14	13	12	11	10	9	8
RESERVED	ERROR_TAG_MATCH	DATA_TAG_MATCH	PING_TAG_MATCH	DATA_FRAME	FRAME_OVERRUN	PING_FRAME	ERR_FRAME
R-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
BUF_UNDERRUN	FRAME_DONE	BUF_OVERRUN	EOF_ERR	TYPE_ERR	CRC_ERR	FRAME_WD_TO	PING_WD_TO
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 36-52. RX_EVT_FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	ERROR_TAG_MATCH	W	0h	Error Tag Match Flag Force bit This bit will cause the corresponding bit in the RX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding bit in the RX_EVT_STS Register. Reset type: SYSRSn
13	DATA_TAG_MATCH	W	0h	Data Tag Match Flag Force bit This bit will cause the corresponding bit in the RX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding bit in the RX_EVT_STS Register. Reset type: SYSRSn
12	PING_TAG_MATCH	W	0h	Ping Tag Match Flag Force bit This bit will cause the corresponding bit in the RX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding bit in the RX_EVT_STS Register. Reset type: SYSRSn
11	DATA_FRAME	W	0h	Data Frame Received Flag Force bit This bit will cause the corresponding bit in the RX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding bit in the RX_EVT_STS Register. Reset type: SYSRSn
10	FRAME_OVERRUN	W	0h	Frame Overrun Flag Force bit This bit will cause the corresponding bit in the RX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding bit in the RX_EVT_STS Register. Reset type: SYSRSn

Table 36-52. RX_EVT_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	PING_FRAME	W	0h	Ping Frame Received Flag Force bit This bit will cause the corresponding bit in the RX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding bit in the RX_EVT_STS Register. Reset type: SYSRSn
8	ERR_FRAME	W	0h	Error Frame Received Flag Force bit This bit will cause the corresponding bit in the RX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding bit in the RX_EVT_STS Register. Reset type: SYSRSn
7	BUF_UNDERRUN	W	0h	Receive Buffer Underrun Flag Force bit This bit will cause the corresponding bit in the RX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding bit in the RX_EVT_STS Register. Reset type: SYSRSn
6	FRAME_DONE	W	0h	Frame Done Flag Force bit This bit will cause the corresponding bit in the RX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding bit in the RX_EVT_STS Register. Reset type: SYSRSn
5	BUF_OVERRUN	W	0h	Receive Buffer Overrun Flag Force bit This bit will cause the corresponding bit in the RX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding bit in the RX_EVT_STS Register. Reset type: SYSRSn
4	EOF_ERR	W	0h	End-of-Frame Error Flag Force bit This bit will cause the corresponding bit in the RX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding bit in the RX_EVT_STS Register. Reset type: SYSRSn
3	TYPE_ERR	W	0h	Frame Type Error Flag Force bit This bit will cause the corresponding bit in the RX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding bit in the RX_EVT_STS Register. Reset type: SYSRSn
2	CRC_ERR	W	0h	CRC Error Flag Force bit This bit will cause the corresponding bit in the RX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding bit in the RX_EVT_STS Register. Reset type: SYSRSn

Table 36-52. RX_EVT_FRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	FRAME_WD_TO	W	0h	Frame Watchdog Timeout Flag Force bit This bit will cause the corresponding bit in the RX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding bit in the RX_EVT_STS Register. Reset type: SYSRSn
0	PING_WD_TO	W	0h	Ping Watchdog Timeout Flag Force bit This bit will cause the corresponding bit in the RX_EVT_STS register to get set. The purpose of this register is to allow software to simulate the effect of the event and test the associated software/ISR. 0h (W) = Writing a 0 to this bit will have no effect. 1h (W) = Force the corresponding bit in the RX_EVT_STS Register. Reset type: SYSRSn

36.6.3.10 RX_BUF_PTR_LOAD Register (Offset = 1Ch) [Reset = 0000h]

RX_BUF_PTR_LOAD is shown in Figure 36-46 and described in Table 36-53.

Return to the Summary Table.

Receive buffer pointer load register

Figure 36-46. RX_BUF_PTR_LOAD Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				BUF_PTR_LOAD			
R-0h				R/W-0h			

Table 36-53. RX_BUF_PTR_LOAD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-0	BUF_PTR_LOAD	R/W	0h	Buffer Pointer Load. This is the value to be loaded into the receive word pointer when written. This is to allow software to force the receiver to start storing the received data starting at a specific location in the buffer. NOTE: The value of the CURR_BUF_PTR in the RX_BUF_PTR_STS will not get reflected immediately. This will take effect only when there is a valid receive operation with incoming clocks after (3 RXCLK + 3 SYCLK) cycles. Reset type: SYSRSn

36.6.3.11 RX_BUF_PTR_STS Register (Offset = 1Eh) [Reset = 0000h]

RX_BUF_PTR_STS is shown in Figure 36-47 and described in Table 36-54.

Return to the Summary Table.

Receive buffer pointer status register

Figure 36-47. RX_BUF_PTR_STS Register

15	14	13	12	11	10	9	8
RESERVED			CURR_WORD_CNT				
R-0h			R-0h				
7	6	5	4	3	2	1	0
RESERVED				CURR_BUF_PTR			
R-0h				R-0h			

Table 36-54. RX_BUF_PTR_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12-8	CURR_WORD_CNT	R	0h	Words Available in the Receive Buffer This bitfield indicates the number of valid data words present in the receive buffer that have not been read by the application software. This bitfield is only valid when there is no active transfer. Note: This value will not be valid if there has been a buffer overrun or underrun condition. Reset type: SYSRSn
7-4	RESERVED	R	0h	Reserved
3-0	CURR_BUF_PTR	R	0h	Current Buffer Pointer Index This bitfield will show the current index of the buffer pointer. This value is only valid when there is no active transmission. Reset type: SYSRSn

36.6.3.12 RX_FRAME_WD_CTRL Register (Offset = 20h) [Reset = 0000h]

RX_FRAME_WD_CTRL is shown in Figure 36-48 and described in Table 36-55.

Return to the Summary Table.

Receive frame watchdog control register

Figure 36-48. RX_FRAME_WD_CTRL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						FRAME_WD_EN	FRAME_WD_CNT_RST
R-0h						R/W-0h	R/W-0h

Table 36-55. RX_FRAME_WD_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-2	RESERVED	R	0h	Reserved
1	FRAME_WD_EN	R/W	0h	Frame Watchdog Counter Enable bit This bit will enable or disable the frame watchdog counter. The counter (RX_FRAME_WD_CNT) will begin counting from 0 when a valid start-of-frame pattern is received. When the reference value (RX_FRAME_WD_REF) is reached, it will generate a frame watchdog timeout event (RX_EVT_STS.FRAME_WD_TO) and the counter value will reset to 0 and continue counting on the next valid start-of-frame. 0h (R/W) = The frame watchdog counter is disabled and not running. 1h (R/W) = The frame watchdog counter logic is enabled and running. Reset type: SYSRSn
0	FRAME_WD_CNT_RST	R/W	0h	Frame Watchdog Counter Reset bit This bit will reset the frame watchdog counter to 0. Writing a 1 to this bit will reset the frame watchdog counter to 0. The counter will stay in reset as long as this bit is set to 1. This bit needs to be cleared to 0 to use the counter 0h (R/W) = Clear the FRAME_WD_CNT_RST. 1h (W) = The frame watchdog counter will be reset to 0. Reset type: SYSRSn

36.6.3.13 RX_FRAME_WD_REF Register (Offset = 24h) [Reset = 00000000h]

RX_FRAME_WD_REF is shown in Figure 36-49 and described in Table 36-56.

Return to the Summary Table.

Receive frame watchdog counter reference

Figure 36-49. RX_FRAME_WD_REF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRAME_WD_REF																															
R/W-0h																															

Table 36-56. RX_FRAME_WD_REF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FRAME_WD_REF	R/W	0h	Frame Watchdog Counter Reference Value This is the 32-bit reference value for the frame watchdog timeout counter. The counter will count up starting from 0 at a valid start-of-frame pattern and continue counting until this value is reached. Reset type: SYSRSn

36.6.3.14 RX_FRAME_WD_CNT Register (Offset = 28h) [Reset = 00000000h]

RX_FRAME_WD_CNT is shown in Figure 36-50 and described in Table 36-57.

Return to the Summary Table.

Receive frame watchdog current count

Figure 36-50. RX_FRAME_WD_CNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRAME_WD_CNT																															
R-0h																															

Table 36-57. RX_FRAME_WD_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FRAME_WD_CNT	R	0h	Frame Watchdog Counter Value This is the 32-bit read-only register which shows the current value of the frame watchdog counter. This counter is reset to 0 in a variety of ways: A write to FRME_WD_CNT_RST, a match with FRAME_WD_REF, or the reception of a successful data frame. Reset type: SYSRSn

36.6.3.15 RX_PING_WD_CTRL Register (Offset = 2Ch) [Reset = 0000h]

RX_PING_WD_CTRL is shown in Figure 36-51 and described in Table 36-58.

Return to the Summary Table.

Receive ping watchdog control register

Figure 36-51. RX_PING_WD_CTRL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						PING_WD_EN	PING_WD_RST
R-0h						R/W-0h	R/W-0h

Table 36-58. RX_PING_WD_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-2	RESERVED	R	0h	Reserved
1	PING_WD_EN	R/W	0h	Ping Watchdog Counter Enable bit This bit will enable or disable the ping watchdog counter. The counter (RX_PING_WD_CNT) will begin counting from 0 when it is enabled. When the reference value (RX_PING_WD_REF) is reached, it will generate a ping watchdog timeout event (RX_EVT_STS.PING_WD_TO) and the counter value will reset to 0, and resume counting 0h (R/W) = The ping watchdog counter is disabled and not running. 1h (R/W) = The ping watchdog counter logic is enabled and running. Reset type: SYSRSn
0	PING_WD_RST	R/W	0h	Ping Watchdog Counter Reset bit This bit will reset the ping watchdog counter to 0. Writing a 1 to this bit will reset the ping watchdog counter to 0. The counter will stay in reset as long as this bit is set to 1. This bit needs to be cleared to 0 to use the counter 0h (R/W) = Clear the PING_WD_RST. 1h (W) = The ping watchdog counter will be reset to 0. Reset type: SYSRSn

36.6.3.16 RX_PING_TAG Register (Offset = 2Eh) [Reset = 0000h]

RX_PING_TAG is shown in Figure 36-52 and described in Table 36-59.

Return to the Summary Table.

Receive ping tag register

Figure 36-52. RX_PING_TAG Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			PING_TAG				RESERVED
R-0h			R-0h				R-0h

Table 36-59. RX_PING_TAG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R	0h	Reserved
4-1	PING_TAG	R	0h	Received Ping Frame Tag This field contains the 4-bit frame tag from the last successfully received ping frame. This is intentionally shifted into bits 4:1 so that the register can be used as a 32-bit address index based on the received tag. Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

36.6.3.17 RX_PING_WD_REF Register (Offset = 30h) [Reset = 00000000h]

RX_PING_WD_REF is shown in Figure 36-53 and described in Table 36-60.

Return to the Summary Table.

Receive ping watchdog counter reference

Figure 36-53. RX_PING_WD_REF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PING_WD_REF																															
R/W-0h																															

Table 36-60. RX_PING_WD_REF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PING_WD_REF	R/W	0h	Ping Watchdog Counter Reference Value This is the 32-bit reference value for the ping watchdog timeout counter. The counter will count up starting from 0 and continue counting until this value is reached. Reset type: SYSRSn

36.6.3.18 RX_PING_WD_CNT Register (Offset = 34h) [Reset = 00000000h]

RX_PING_WD_CNT is shown in Figure 36-54 and described in Table 36-61.

Return to the Summary Table.

Receive pingwatchdog current count

Figure 36-54. RX_PING_WD_CNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PING_WD_CNT																															
R-0h																															

Table 36-61. RX_PING_WD_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PING_WD_CNT	R	0h	Ping Watchdog Counter Value This is the 32-bit read-only register which shows the current value of the ping watchdog counter. This counter is reset to 0 in a variety of ways: A write to PING_WD_RST, a match with PING_WD_REF, or the reception of a ping frame. Reset type: SYSRSn

36.6.3.19 RX_INT1_CTRL Register (Offset = 38h) [Reset = 0000h]

RX_INT1_CTRL is shown in Figure 36-55 and described in Table 36-62.

Return to the Summary Table.

Receive interrupt control register for RX_INT1

Figure 36-55. RX_INT1_CTRL Register

15	14	13	12	11	10	9	8
RESERVED	INT1_EN_ERROR_TAG_MATCH	INT1_EN_DATA_TAG_MATCH	INT1_EN_PING_TAG_MATCH	INT1_EN_DATA_FRAME	INT1_EN_FRAME_OVERRUN	INT1_EN_PING_FRAME	INT1_EN_ERR_FRAME
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INT1_EN_UNDERRUN	INT1_EN_FRAME_DONE	INT1_EN_OVERRUN	INT1_EN_EOF_ERR	INT1_EN_TYPE_ERR	INT1_EN_CRC_ERR	INT1_EN_FRAME_WD_TO	INT1_EN_PING_WD_TO
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 36-62. RX_INT1_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	INT1_EN_ERROR_TAG_MATCH	R/W	0h	Enable Error Frame Received with Tag Match Interrupt to INT1 bit This is an enable register which decides whether an interrupt (RX_INT1) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT1. 1h (R/W) = An error frame received with matching tag will trigger an interrupt on RX_INT1. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
13	INT1_EN_DATA_TAG_MATCH	R/W	0h	Enable Data Frame Received with Tag Match Interrupt to INT1 bit This is an enable register which decides whether an interrupt (RX_INT1) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT1. 1h (R/W) = A data frame received with matching tag will trigger an interrupt on RX_INT1. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
12	INT1_EN_PING_TAG_MATCH	R/W	0h	Enable Ping Frame Received with Tag Match Interrupt to INT1 bit This is an enable register which decides whether an interrupt (RX_INT1) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT1. 1h (R/W) = A ping frame received with matching tag will trigger an interrupt on RX_INT1. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
11	INT1_EN_DATA_FRAME	R/W	0h	Enable Data Frame Received Interrupt to INT1 bit This is an enable register which decides whether an interrupt (RX_INT1) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT1. 1h (R/W) = A data frame received event will trigger an interrupt on RX_INT1. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn

Table 36-62. RX_INT1_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INT1_EN_FRAME_OVERRUN	R/W	0h	Enable Frame Overrun Interrupt to INT1 bit This is an enable register which decides whether an interrupt (RX_INT1) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT1. 1h (R/W) = A frame overrun event will trigger an interrupt on RX_INT1. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
9	INT1_EN_PING_FRAME	R/W	0h	Enable Ping Frame Received Interrupt to INT1 bit This is an enable register which decides whether an interrupt (RX_INT1) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT1. 1h (R/W) = A ping frame received event will trigger an interrupt on RX_INT1. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
8	INT1_EN_ERR_FRAME	R/W	0h	Enable ERROR Frame Received Interrupt to INT1 bit This is an enable register which decides whether an interrupt (RX_INT1) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT1. 1h (R/W) = A error frame received event will trigger an interrupt on RX_INT1. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
7	INT1_EN_UNDERRUN	R/W	0h	Enable Buffer Underrun Interrupt to INT1 bit This is an enable register which decides whether an interrupt (RX_INT1) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT1. 1h (R/W) = A buffer underrun event will trigger an interrupt on RX_INT1. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
6	INT1_EN_FRAME_DONE	R/W	0h	Enable Frame Done Interrupt to INT1 bit This is an enable register which decides whether an interrupt (RX_INT1) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT1. 1h (R/W) = A frame done event will trigger an interrupt on RX_INT1. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
5	INT1_EN_OVERRUN	R/W	0h	Enable Receive Buffer Overrun Interrupt to INT1 bit This is an enable register which decides whether an interrupt (RX_INT1) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT1. 1h (R/W) = A receive buffer overrun event will trigger an interrupt on RX_INT1. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
4	INT1_EN_EOF_ERR	R/W	0h	Enable End-of-Frame Error Interrupt to INT1 bit This is an enable register which decides whether an interrupt (RX_INT1) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT1. 1h (R/W) = An end-of-frame error event will trigger an interrupt on RX_INT1. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn

Table 36-62. RX_INT1_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	INT1_EN_TYPE_ERR	R/W	0h	Enable Frame Type Error Interrupt to INT1 bit This is an enable register which decides whether an interrupt (RX_INT1) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT1. 1h (R/W) = A frame type error event will trigger an interrupt on RX_INT1. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
2	INT1_EN_CRC_ERR	R/W	0h	Enable CRC Error Interrupt to INT1 bit This is an enable register which decides whether an interrupt (RX_INT1) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT1. 1h (R/W) = A CRC error will trigger an interrupt on RX_INT1. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
1	INT1_EN_FRAME_WD_TO	R/W	0h	Enable Frame Watchdog Timeout Interrupt to INT1 bit This is an enable register which decides whether an interrupt (RX_INT1) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT1. 1h (R/W) = A frame watchdog timeout event will trigger an interrupt on RX_INT1. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
0	INT1_EN_PING_WD_TO	R/W	0h	Enable Ping Watchdog Timeout Interrupt to INT1 bit This is an enable register which decides whether an interrupt (RX_INT1) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT1. 1h (R/W) = A ping watchdog timeout event will trigger an interrupt on RX_INT1. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn

36.6.3.20 RX_INT2_CTRL Register (Offset = 3Ah) [Reset = 0000h]

RX_INT2_CTRL is shown in Figure 36-56 and described in Table 36-63.

Return to the Summary Table.

Receive interrupt control register for RX_INT2

Figure 36-56. RX_INT2_CTRL Register

15	14	13	12	11	10	9	8
RESERVED	INT2_EN_ERROR_TAG_MATCH	INT2_EN_DATA_TAG_MATCH	INT2_EN_PING_TAG_MATCH	INT2_EN_DATA_FRAME	INT2_EN_FRAME_OVERRUN	INT2_EN_PING_FRAME	INT2_EN_ERR_FRAME
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INT2_EN_UNDERRUN	INT2_EN_FRAME_DONE	INT2_EN_OVERRUN	INT2_EN_EOF_ERR	INT2_EN_TYPE_ERR	INT2_EN_CRC_ERR	INT2_EN_FRAME_WD_TO	INT2_EN_PING_WD_TO
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 36-63. RX_INT2_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	INT2_EN_ERROR_TAG_MATCH	R/W	0h	Enable Error Frame Received with Tag Match Interrupt to INT2 bit This is an enable register which decides whether an interrupt (RX_INT2) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT2. 1h (R/W) = An error frame received with matching tag will trigger an interrupt on RX_INT2. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
13	INT2_EN_DATA_TAG_MATCH	R/W	0h	Enable Data Frame Received with Tag Match Interrupt to INT2 bit This is an enable register which decides whether an interrupt (RX_INT2) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT2. 1h (R/W) = A data frame received with matching tag will trigger an interrupt on RX_INT2. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
12	INT2_EN_PING_TAG_MATCH	R/W	0h	Enable Ping Frame Received with Tag Match Interrupt to INT2 bit This is an enable register which decides whether an interrupt (RX_INT2) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT2. 1h (R/W) = A ping frame received with matching tag will trigger an interrupt on RX_INT2. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
11	INT2_EN_DATA_FRAME	R/W	0h	Enable Data Frame Received Interrupt to INT2 bit This is an enable register which decides whether an interrupt (RX_INT2) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT2. 1h (R/W) = A data frame received event will trigger an interrupt on RX_INT2. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn

Table 36-63. RX_INT2_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	INT2_EN_FRAME_OVERRUN	R/W	0h	Enable Frame Overrun Interrupt to INT2 bit This is an enable register which decides whether an interrupt (RX_INT2) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT2. 1h (R/W) = A frame overrun event will trigger an interrupt on RX_INT2. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
9	INT2_EN_PING_FRAME	R/W	0h	Enable Ping Frame Received Interrupt to INT2 bit This is an enable register which decides whether an interrupt (RX_INT2) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT2. 1h (R/W) = A ping frame received event will trigger an interrupt on RX_INT2. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
8	INT2_EN_ERR_FRAME	R/W	0h	Enable Error Frame Received Interrupt to INT2 bit This is an enable register which decides whether an interrupt (RX_INT2) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT2. 1h (R/W) = A error frame received event will trigger an interrupt on RX_INT2. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
7	INT2_EN_UNDERRUN	R/W	0h	Enable Buffer Underrun Interrupt to INT2 bit This is an enable register which decides whether an interrupt (RX_INT2) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT2. 1h (R/W) = A buffer underrun event will trigger an interrupt on RX_INT2. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
6	INT2_EN_FRAME_DONE	R/W	0h	Enable Frame Done Interrupt to INT2 bit This is an enable register which decides whether an interrupt (RX_INT2) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT2. 1h (R/W) = A frame done event will trigger an interrupt on RX_INT2. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
5	INT2_EN_OVERRUN	R/W	0h	Enable Buffer Overrun Interrupt to INT2 bit This is an enable register which decides whether an interrupt (RX_INT2) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT2. 1h (R/W) = A buffer overrun event will trigger an interrupt on RX_INT2. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
4	INT2_EN_EOF_ERR	R/W	0h	Enable End-of-Frame Error Interrupt to INT2 bit This is an enable register which decides whether an interrupt (RX_INT2) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT2. 1h (R/W) = An end-of-frame error event will trigger an interrupt on RX_INT2. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn

Table 36-63. RX_INT2_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	INT2_EN_TYPE_ERR	R/W	0h	Enable Frame Type Error Interrupt to INT2 bit This is an enable register which decides whether an interrupt (RX_INT2) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT2. 1h (R/W) = A frame type error event will trigger an interrupt on RX_INT2. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
2	INT2_EN_CRC_ERR	R/W	0h	Enable CRC Error Interrupt to INT2 bit This is an enable register which decides whether an interrupt (RX_INT2) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT2. 1h (R/W) = A CRC error will trigger an interrupt on RX_INT2. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
1	INT2_EN_FRAME_WD_TO	R/W	0h	Enable Frame Watchdog Timeout Interrupt to INT2 bit This is an enable register which decides whether an interrupt (RX_INT2) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT2. 1h (R/W) = A frame watchdog timeout event will trigger an interrupt on RX_INT2. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn
0	INT2_EN_PING_WD_TO	R/W	0h	Enable Ping Watchdog Timeout Interrupt to INT2 bit This is an enable register which decides whether an interrupt (RX_INT2) will be generated on the enabled event. 0h (R/W) = This event will not trigger an interrupt on RX_INT2. 1h (R/W) = A ping watchdog timeout event will trigger an interrupt on RX_INT2. The event itself will be latched in the corresponding bit in the RX_EVT_STS Register Reset type: SYSRSn

36.6.3.21 RX_LOCK_CTRL Register (Offset = 3Ch) [Reset = 0000h]

RX_LOCK_CTRL is shown in Figure 36-57 and described in Table 36-64.

Return to the Summary Table.

Receive lock control register

Figure 36-57. RX_LOCK_CTRL Register

15	14	13	12	11	10	9	8
KEY							
W-0h							
7	6	5	4	3	2	1	0
RESERVED							LOCK
R-0h							R/W-0h

Table 36-64. RX_LOCK_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	KEY	W	0h	Write Key. In order to write to this register, 0xA5 must be written to this field at the same time. Otherwise, writes are ignored. The key is cleared immediately after writing, so it must be written again for every change to this register. Reset type: SYSRSn
7-1	RESERVED	R	0h	Reserved
0	LOCK	R/W	0h	Control Register Lock Enable bit This bit locks the contents of all the receive control registers that support a lock protection. Once locked, further writes will not take effect until SYSRS unlocks the register. Once set, further writes even to this bit will be ignored. 0h (R/W) = Receive control registers can be modified and are not locked. 1h (R/W) = Receive control registers are locked and cannot be modified until this bit is cleared by SYSRS. Any further writes to this bit are ignored. Note: The KEY field must contatin 0xA5 for any write to this bit to take effect. Reset type: SYSRSn

36.6.3.22 RX_ECC_DATA Register (Offset = 40h) [Reset = 00000000h]

RX_ECC_DATA is shown in Figure 36-58 and described in Table 36-65.

Return to the Summary Table.

Receive ECC data register

Figure 36-58. RX_ECC_DATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_HIGH																DATA_LOW															
R/W-0h																R/W-0h															

Table 36-65. RX_ECC_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DATA_HIGH	R/W	0h	Upper 16 bits of ECC Data Writing to this bitfield will cause the ECC logic to compute the ECC(SEC-DED) the entire 32-bit register and update TX_ECC_VAL register with the results. Software should write to these 16 bits of the register in a 32-bit write when needing to compute ECC for 32-bits for the full TX_ECC_DATA register. Reset type: SYSRSn
15-0	DATA_LOW	R/W	0h	Lower 16 bits of ECC Data Writing to this bitfield will cause the ECC logic to compute the ECC(SEC-DED) for these 16 bits and update the TX_ECC_VAL register with the results. Software should write to these register bits as a 16-bit write when needing to compute ECC for 16-bits. Reset type: SYSRSn

36.6.3.23 RX_ECC_VAL Register (Offset = 44h) [Reset = 0000h]

RX_ECC_VAL is shown in Figure 36-59 and described in Table 36-66.

Return to the Summary Table.

Receive ECC value register

Figure 36-59. RX_ECC_VAL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	ECC_VAL						
R-0h	R/W-0h						

Table 36-66. RX_ECC_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6-0	ECC_VAL	R/W	0h	ECC Value for SEC-DED check This field contains the ECC value to be used for SEC-DED either for 16-bit or 32-bit data in the RX_ECC_DATA register. Reset type: SYSRSn

36.6.3.24 RX_ECC_SEC_DATA Register (Offset = 48h) [Reset = 00000000h]

RX_ECC_SEC_DATA is shown in Figure 36-60 and described in Table 36-67.

Return to the Summary Table.

Receive ECC corrected data register

Figure 36-60. RX_ECC_SEC_DATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_DATA																															
R-0h																															

Table 36-67. RX_ECC_SEC_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SEC_DATA	R	0h	ECC Single Error Corrected Data The ECC corrected data will be available in this register. This value is valid only when there are no bit errors, or a single bit error was detected. Otherwise, the contents of this register are invalid and should not be used. Reset type: SYSRSn

36.6.3.25 RX_ECC_LOG Register (Offset = 4Ch) [Reset = 0003h]

RX_ECC_LOG is shown in Figure 36-61 and described in Table 36-68.

Return to the Summary Table.

Receive ECC log and status register

Figure 36-61. RX_ECC_LOG Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						MBE	SBE
R-0h						R-1h	R-1h

Table 36-68. RX_ECC_LOG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-2	RESERVED	R	0h	Reserved
1	MBE	R	1h	Multiple Bit Errors Detected This bit indicates the occurrence of multiple bit errors.The data is corrupted and cannot be corrected. If this bit is set, the data present in RX_ECC_SEC_DATA is invalid and should not be used. 0h (R) Multiple Bit Errors were not detected. Check the SBE bit for single bit errors. 1h (R) Multiple Bit Errors were detected. The data is not able to be corrected. The value present in RX_ECC_SEC_DATA is invalid and should not be used. Reset type: SYSRSn
0	SBE	R	1h	Single Bit Error Detected This bit indicates the occurrence of a single bit error in the data. The data is autocorrected and placed into the RX_ECC_SEC_DATA register. This bit is valid only if MBE is 0. 0h (R) No bit errors were detected. The value in RX_ECC_SEC_DATA is correct. 1h (R) A single bit error was detected and corrected. The corrected data is present in RX_ECC_SEC_DATA. Reset type: SYSRSn

36.6.3.26 RX_FRAME_TAG_CMP Register (Offset = 50h) [Reset = 0000h]

RX_FRAME_TAG_CMP is shown in Figure 36-62 and described in Table 36-69.

Return to the Summary Table.

Receive frame tag compare register

Figure 36-62. RX_FRAME_TAG_CMP Register

15	14	13	12	11	10	9	8
RESERVED						BROADCAST_EN	CMP_EN
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TAG_MASK				TAG_REF			
R/W-0h				R/W-0h			

Table 36-69. RX_FRAME_TAG_CMP Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9	BROADCAST_EN	R/W	0h	Broadcast Enable bit This will enable the reception of a ping frame broadcast. When this bit is set, bit 3 of the received tag will be treated as a broadcast notification. If bit 3 of the received tag is set to 1, a ping tag match event will be triggered regardless of the. A match caused by the comparison of TAG_MASK and TAG_REF will still be considered a match and the frame tag match event will be triggered as normal This bit only takes effect only if CMP_EN is set to 1. 0h (R/W) Broadcast frame match disabled. 1h (R/W) Broadcast frame match enabled. Reset type: SYSRSn
8	CMP_EN	R/W	0h	Frame Tag Compare Enable bit Set this bit to enable the comparison of an incoming frame tag and the value stored in the frame tag reference. A match caused by the comparison of TAG_MASK, TAG_REF, and the incoming frame tag will trigger the apprpriate frame tag match event. 0h (R/W) Frame tag comparison is disabled. 1h (R/W) Frame tag comparison is enabled. Reset type: SYSRSn
7-4	TAG_MASK	R/W	0h	Frame Tag Mask Any bit position in this register set to 0 will be used in the comparison of the incoming frame tag and the value stored in TAG_REF. A bit position set to 1 will be ignored in the tag comparison. This mask value is used only for non-ping frames. Reset type: SYSRSn
3-0	TAG_REF	R/W	0h	Frame Tag Reference The reference tag to check against when comparing the TAG_MASK and the incoming frame tag. This reference value is used only for non-ping frames. Reset type: SYSRSn

36.6.3.27 RX_PING_TAG_CMP Register (Offset = 52h) [Reset = 0000h]

RX_PING_TAG_CMP is shown in Figure 36-63 and described in Table 36-70.

Return to the Summary Table.

Receive ping tag compare register

Figure 36-63. RX_PING_TAG_CMP Register

15	14	13	12	11	10	9	8
RESERVED						BROADCAST_EN	CMP_EN
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TAG_MASK				TAG_REF			
R/W-0h				R/W-0h			

Table 36-70. RX_PING_TAG_CMP Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9	BROADCAST_EN	R/W	0h	Broadcast Enable bit This will enable the reception of a ping frame broadcast. When this bit is set, bit 3 of the received tag will be treated as a broadcast notification. If bit 3 of the received tag is set to 1, a ping tag match event will be triggered regardless of the. A match caused by the comparison of TAG_MASK and TAG_REF will still be considered a match and the ping tag match event will be triggered as normal This bit only takes effect only if CMP_EN is set to 1. 0h (R/W) Broadcast frame match disabled. 1h (R/W) Broadcast frame match enabled. Reset type: SYSRSn
8	CMP_EN	R/W	0h	Ping Tag Compare Enable bit Set this bit to enable the comparison of an incoming ping tag and the value stored in the ping tag reference. A match caused by the comparison of TAG_MASK, TAG_REF, and the incoming ping tag will trigger a ping frame tag match event. 0h (R/W) Ping tag comparison is disabled. 1h (R/W) Ping tag comparison is enabled. Reset type: SYSRSn
7-4	TAG_MASK	R/W	0h	Ping Tag Mask Any bit position in this register set to 0 will be used in the comparison of the incoming ping frame tag and the value stored in TAG_REF. A bit position set to 1 will be ignored in the tag comparison. This mask value is used only for ping frames. Reset type: SYSRSn
3-0	TAG_REF	R/W	0h	Ping Tag Reference The reference tag to check against when comparing the TAG_MASK and the incoming ping tag. This reference value is used only for ping frames. Reset type: SYSRSn

36.6.3.28 RX_TRIG_CTRL_0 Register (Offset = 58h) [Reset = 00000000h]

RX_TRIG_CTRL_0 is shown in Figure 36-64 and described in Table 36-71.

Return to the Summary Table.

Receive Trigger Control register 0

Figure 36-64. RX_TRIG_CTRL_0 Register

31	30	29	28	27	26	25	24
RX_TRIG_DLY							
R/W-0h							
23	22	21	20	19	18	17	16
RX_TRIG_DLY							
R/W-0h							
15	14	13	12	11	10	9	8
RX_TRIG_DLY							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED			TRIG_SEL				TRIG_EN
R-0h			R/W-0h				R/W-0h

Table 36-71. RX_TRIG_CTRL_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RX_TRIG_DLY	R/W	0h	This is the 24 bit count of the trigger delay in SYSCLK cycles. If enabled, the Trigger-1 output of the trigger module will generate a 3 SYSCLK wide trigger pulse after the selected input trigger source sees a rising edge with a delay defined by this 24-bt value. Reset type: SYSRSn
7-5	RESERVED	R	0h	Reserved
4-1	TRIG_SEL	R/W	0h	This is the mux select value which selects which of the inputs will be used as the trigger source. Reset type: SYSRSn
0	TRIG_EN	R/W	0h	This is the enable for the RX output trigger generation. The output triggers will be generated only if this bit is set to 1. If this bit is 0, then no trigger will be generated by this module. Reset type: SYSRSn

36.6.3.29 RX_TRIG_WIDTH_0 Register (Offset = 5Ch) [Reset = 00000000h]

RX_TRIG_WIDTH_0 is shown in Figure 36-65 and described in Table 36-72.

Return to the Summary Table.

Receive Trigger Wdith register 0

Figure 36-65. RX_TRIG_WIDTH_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_TRIG_WIDTH															
R-0h																R/W-0h															

Table 36-72. RX_TRIG_WIDTH_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	RX_TRIG_WIDTH	R/W	0h	This register decides the width(in SYSCLK cycles) of wide pulse output of the RX trigger module. Reset type: SYSRSn

36.6.3.30 RX_DLYLINE_CTRL Register (Offset = 60h) [Reset = 0000h]

RX_DLYLINE_CTRL is shown in Figure 36-66 and described in Table 36-73.

Return to the Summary Table.

Receive delay line control register

Figure 36-66. RX_DLYLINE_CTRL Register

15	14	13	12	11	10	9	8
RESERVED	RXD1_DLY					RXD0_DLY	
R-0h	R/W-0h					R/W-0h	
7	6	5	4	3	2	1	0
RXD0_DLY				RXCLK_DLY			
R/W-0h				R/W-0h			

Table 36-73. RX_DLYLINE_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14-10	RXD1_DLY	R/W	0h	Delay Line Tap Select for RXD1 This bitfield selects the number of delay elements inserted into the RXD1 path from the pin boundary to the receiver core. 0h (R/W) Zero delay elements are included in the RXD1 path. RXD1 is taken directly from the pin. 1h (R/W) One delay element is included in the RXD1 path. 2h (R/W) Two delay elements are included in the RXD1 path. ... 1Fh (R/W) 31 delay elements are included in the RXD1 path, the maximum. Reset type: SYSRSn
9-5	RXD0_DLY	R/W	0h	Delay Line Tap Select for RXD0 This bitfield selects the number of delay elements inserted into the RXD0 path from the pin boundary to the receiver core. 0h (R/W) Zero delay elements are included in the RXD0 path. RXD0 is taken directly from the pin. 1h (R/W) One delay element is included in the RXD0 path. 2h (R/W) Two delay elements are included in the RXD0 path. ... 1Fh (R/W) 31 delay elements are included in the RXD0 path, the maximum. Reset type: SYSRSn
4-0	RXCLK_DLY	R/W	0h	Delay Line Tap Select for RXCLK This bitfield selects the number of delay elements inserted into the RXCLK path from the pin boundary to the receiver core. 0h (R/W) Zero delay elements are included in the RXCLK path. RXCLK is taken directly from the pin. 1h (R/W) One delay element is included in the RXCLK path. 2h (R/W) Two delay elements are included in the RXCLK path. ... 1Fh (R/W) 31 delay elements are included in the RXCLK path, the maximum. Reset type: SYSRSn

36.6.3.31 RX_TRIG_CTRL_1 Register (Offset = 64h) [Reset = 00000000h]

RX_TRIG_CTRL_1 is shown in Figure 36-67 and described in Table 36-74.

Return to the Summary Table.

Receive Trigger Control register 1

Figure 36-67. RX_TRIG_CTRL_1 Register

31	30	29	28	27	26	25	24
RX_TRIG_DLY							
R/W-0h							
23	22	21	20	19	18	17	16
RX_TRIG_DLY							
R/W-0h							
15	14	13	12	11	10	9	8
RX_TRIG_DLY							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED			TRIG_SEL				TRIG_EN
R-0h			R/W-0h				R/W-0h

Table 36-74. RX_TRIG_CTRL_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RX_TRIG_DLY	R/W	0h	This is the 24 bit count of the trigger delay in SYSCLK cycles. If enabled, the Trigger-1 output of the trigger module will generate a 3 SYSCLK wide trigger pulse after the selected input trigger source sees a rising edge with a delay defined by this 24-bt value. Reset type: SYSRSn
7-5	RESERVED	R	0h	Reserved
4-1	TRIG_SEL	R/W	0h	This is the mux select value which selects which of the inputs will be used as the trigger source. Reset type: SYSRSn
0	TRIG_EN	R/W	0h	This is the enable for the RX output trigger generation. The output triggers will be generated only if this bit is set to 1. If this bit is 0, then no trigger will be generated by this module. Reset type: SYSRSn

36.6.3.32 RX_TRIG_CTRL_2 Register (Offset = 68h) [Reset = 00000000h]

RX_TRIG_CTRL_2 is shown in Figure 36-68 and described in Table 36-75.

Return to the Summary Table.

Receive Trigger Control register 2

Figure 36-68. RX_TRIG_CTRL_2 Register

31	30	29	28	27	26	25	24
RX_TRIG_DLY							
R/W-0h							
23	22	21	20	19	18	17	16
RX_TRIG_DLY							
R/W-0h							
15	14	13	12	11	10	9	8
RX_TRIG_DLY							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED			TRIG_SEL				TRIG_EN
R-0h			R/W-0h				R/W-0h

Table 36-75. RX_TRIG_CTRL_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RX_TRIG_DLY	R/W	0h	This is the 24 bit count of the trigger delay in SYSCLK cycles. If enabled, the Trigger-1 output of the trigger module will generate a 3 SYSCLK wide trigger pulse after the selected input trigger source sees a rising edge with a delay defined by this 24-bt value. Reset type: SYSRSn
7-5	RESERVED	R	0h	Reserved
4-1	TRIG_SEL	R/W	0h	This is the mux select value which selects which of the inputs will be used as the trigger source. Reset type: SYSRSn
0	TRIG_EN	R/W	0h	This is the enable for the RX output trigger generation. The output triggers will be generated only if this bit is set to 1. If this bit is 0, then no trigger will be generated by this module. Reset type: SYSRSn

36.6.3.33 RX_TRIG_CTRL_3 Register (Offset = 6Ch) [Reset = 00000000h]

RX_TRIG_CTRL_3 is shown in Figure 36-69 and described in Table 36-76.

Return to the Summary Table.

Receive Trigger Control register 3

Figure 36-69. RX_TRIG_CTRL_3 Register

31	30	29	28	27	26	25	24
RX_TRIG_DLY							
R/W-0h							
23	22	21	20	19	18	17	16
RX_TRIG_DLY							
R/W-0h							
15	14	13	12	11	10	9	8
RX_TRIG_DLY							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED			TRIG_SEL				TRIG_EN
R-0h			R/W-0h				R/W-0h

Table 36-76. RX_TRIG_CTRL_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RX_TRIG_DLY	R/W	0h	This is the 24 bit count of the trigger delay in SYSCLK cycles. If enabled, the Trigger-1 output of the trigger module will generate a 3 SYSCLK wide trigger pulse after the selected input trigger source sees a rising edge with a delay defined by this 24-bt value. Reset type: SYSRSn
7-5	RESERVED	R	0h	Reserved
4-1	TRIG_SEL	R/W	0h	This is the mux select value which selects which of the inputs will be used as the trigger source. Reset type: SYSRSn
0	TRIG_EN	R/W	0h	This is the enable for the RX output trigger generation. The output triggers will be generated only if this bit is set to 1. If this bit is 0, then no trigger will be generated by this module. Reset type: SYSRSn

36.6.3.34 RX_VIS_1 Register (Offset = 70h) [Reset = 00000000h]

RX_VIS_1 is shown in Figure 36-70 and described in Table 36-77.

Return to the Summary Table.

Receive debug visibility register 1

Figure 36-70. RX_VIS_1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RX_CORE_STS	RESERVED		
R-0h				R-0h	R-0h		

Table 36-77. RX_VIS_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3	RX_CORE_STS	R	0h	Receiver Core Status bit This bit indicates the status of the receiver core. If this bit is set, the receiver should undergo a reset and subsequent resynchronization with the transmitter. This bit will be always be set when the receiver has detected and end of frame error or a frame type error. This bit can also be set if the receiver becomes corrupted due to noise on the signal lines. If the receiver has experienced a ping watchdog or frame watchdog timeout, this bit should be read to determine if the cause was due to a corrupt transaction, thus putting the receiver core into an unrecoverable state. Only a soft reset will reset the recevier core and thus reset this bit. 0h (R) The receiver core is operating normally. 1h (R) The receiver core has entered into an error state and should be reset. Reset type: SYSRSn
2-0	RESERVED	R	0h	Reserved

36.6.3.35 RX_UDATA_FILTER Register (Offset = 74h) [Reset = 0000h]

RX_UDATA_FILTER is shown in Figure 36-71 and described in Table 36-78.

Return to the Summary Table.

Receive User Data Filter Control register

Figure 36-71. RX_UDATA_FILTER Register

15	14	13	12	11	10	9	8
UDATA_MASK							
R/W-0h							
7	6	5	4	3	2	1	0
UDATA_REF							
R/W-0h							

Table 36-78. RX_UDATA_FILTER Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	UDATA_MASK	R/W	0h	Bit Mask to be used for comparing the USERDATA field when filtering is enabled. Every bit that is '1' in this register will be masked for comparison. If a bit position is '1', then it will be considered a successful match for that bit position. Reset type: SYSRSn
7-0	UDATA_REF	R/W	0h	Reference to be used for comparing the USERDATA field when filtering is enabled. Reset type: SYSRSn

36.6.3.36 RX_BUF_BASE_y Register (Offset = 80h + formula) [Reset = 0000h]

RX_BUF_BASE_y is shown in Figure 36-72 and described in Table 36-79.

Return to the Summary Table.

Base address for receive data buffer

Offset = 80h + (y * 2h); where y = 0h to Fh

Figure 36-72. RX_BUF_BASE_y Register

15	14	13	12	11	10	9	8
BASE_ADDRESS							
R-0h							
7	6	5	4	3	2	1	0
BASE_ADDRESS							
R-0h							

Table 36-79. RX_BUF_BASE_y Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	BASE_ADDRESS	R	0h	Receive Data Buffer Base Address This is the base address of the 16-word data buffer used by the receiver. Reset type: SYSRSn

Chapter 37

Inter-Integrated Circuit Module (I2C)

This chapter describes the features and operation of the inter-integrated circuit (I2C) module. The I2C module provides an interface between one of these devices and devices compliant with the NXP Semiconductors Inter-IC bus (I2C bus) specification version 2.1, and connected by way of an I2C bus. External components attached to this 2-wire serial bus can transmit/receive 1- to 8-bit data to/from the device through the I2C module. This chapter assumes the reader is familiar with the I2C bus specification.

Note

A unit of data transmitted or received by the I2C module can have fewer than 8 bits; however, for convenience, a unit of data is called a data byte throughout this chapter. The number of bits in a data byte is selectable by way of the BC bits of the mode register, I2CMDR.

37.1 Introduction

The I2C module supports any target or controller I2C-compatible device. Figure 37-1 shows an example of multiple I2C modules connected for a two-way transfer from one device to other devices.

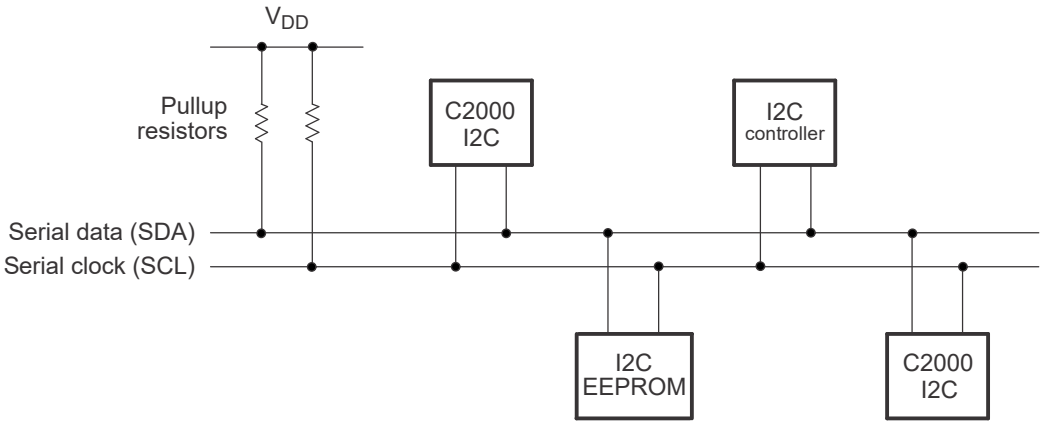


Figure 37-1. Multiple I2C Modules Connected

37.1.1 I2C Related Collateral

Foundational Materials

- C28x Academy - I2C
- C29x Academy - Inter-Integrated Circuit (I2C)
- I2C Hardware Overview (Video)
- I2C Protocol Overview (Video)
- *Understanding the I2C Bus Application Report*

Getting Started Materials

- *Configuring the TMS320F280x DSP as an I2C Processor Application Report*
- I2C Buffers Overview (Video)
- *I2C Dynamic Addressing Application Report*
- I2C translators overview (Video)
- *Interfacing EEPROM Using C2000 I2C Module Application Report*
- *Why, When, and How to use I2C Buffers Application Report*

Expert Materials

- *I2C Bus Pull-Up Resistor Calculation Application Report*
- *Maximum Clock Frequency of I2C Bus Using Repeaters Application Report*

37.1.2 Features

The I2C module has the following features:

- Compliance with the NXP Semiconductors I2C bus specification (version 2.1):
  - Support for 8-bit format transfers
  - 7-bit and 10-bit addressing modes
  - General call
  - START byte mode
  - Support for multiple controller-transmitters and target-receivers
  - Support for multiple target-transmitters and controller-receivers
  - Combined controller transmit/receive and receive/transmit mode
  - Data transfer rate from 10kbps up to 400kbps (Fast-mode)
- Extended Automatic Clock Stretching and Manual Clock Stretching modes
- Receive FIFO and Transmitter FIFO (16-deep x 8-bit FIFO)
- Supports two Interrupt Controller interrupts:
- – I2Cx Interrupt – Any of the following events can be configured to generate an I2Cx interrupt:
    - Transmit-data ready
    - Receive-data ready
    - Register-access ready
    - No-acknowledgment received
    - Arbitration lost
    - Stop condition detected
    - Addressed as target
    - Extended Automatic Clock Stretch
  - I2Cx_FIFO interrupts:
    - Transmit FIFO interrupt
    - Receive FIFO interrupt
- Module enable and disable capability
- Free data format mode

37.1.3 Features Not Supported

The I2C module does not support:

- High-speed mode (Hs-mode)
- CBUS-compatibility mode

37.1.4 Functional Overview

Each device connected to an I2C bus is recognized by a unique address. Each device can operate as either a transmitter or a receiver, depending on the function of the device. A device connected to the I2C bus can also be considered as the controller or the target when performing data transfers. A controller device is the device that initiates a data transfer on the bus and generates the clock signals to permit that transfer. During this transfer, any device addressed by this controller is considered a target. The I2C module supports the multi-controller mode, in which one or more devices capable of controlling an I2C bus can be connected to the same I2C bus.

For data communication, the I2C module has a serial data pin (SDA) and a serial clock pin (SCL), as shown in Figure 37-2. These two pins carry information between the C28x device and other devices connected to the I2C bus. The SDA and SCL pins are both bidirectional and each must be connected to a positive supply voltage using a pull-up resistor. When the bus is free, both pins are high. The driver of these two pins has an open-drain configuration to perform the required wired-AND function.

There are two major transfer techniques:

- Standard Mode: Send exactly n data values, where n is a value you program in an I2C module register. See the I2CCNT register in the *I2C Registers* section for more information.
- Repeat Mode: Keep sending data values until you use software to initiate a STOP condition or a new START condition. See the I2CMDR register in the *I2C Registers* section for RM bit information.

The I2C module consists of the following primary blocks:

- A serial interface: one data pin (SDA) and one clock pin (SCL)
- Data registers and FIFOs to temporarily hold receive data and transmit data traveling between the SDA pin and the CPU
- Control and status registers
- A peripheral bus interface to enable the CPU to access the I2C module registers and FIFOs.
- A clock synchronizer to synchronize the I2C input clock (from the device clock generator) and the clock on the SCL pin, and to synchronize data transfers with controllers of different clock speeds
- A prescaler to divide down the input clock that is driven to the I2C module
- A noise filter on each of the two pins, SDA and SCL
- An arbitrator to handle arbitration between the I2C module (when the I2C module is a controller) and another controller
- Interrupt generation logic, so that an interrupt can be sent to the CPU
- FIFO interrupt generation logic, so that FIFO access can be synchronized to data reception and data transmission in the I2C module

Figure 37-2 shows the four registers used for transmission and reception in non-FIFO mode. The CPU writes data for transmission to I2CDXR and reads received data from I2CDRR. When the I2C module is configured as a transmitter, data written to I2CDXR is copied to I2CXSR and shifted out on the SDA pin one bit at a time. When the I2C module is configured as a receiver, received data is shifted into I2CRSR and then copied to I2CDRR.

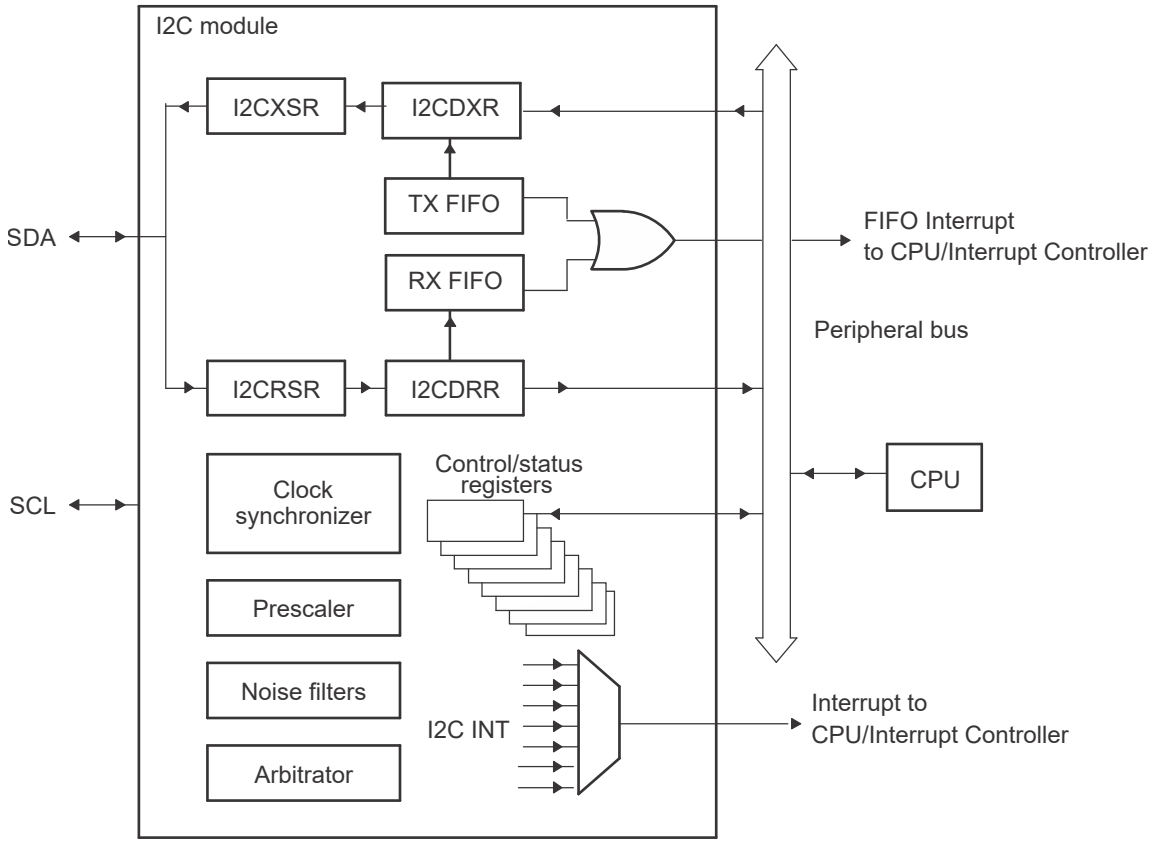


Figure 37-2. I2C Module Conceptual Block Diagram

37.1.5 Clock Generation

The I2C module clock determines the frequency at which the I2C module operates. A programmable prescaler in the I2C module divides down the SYSCLK to produce the I2C module clock and this I2C module clock is divided further to produce the I2C controller clock on the SCL pin. Figure 37-3 shows the clock generation diagram for I2C module.

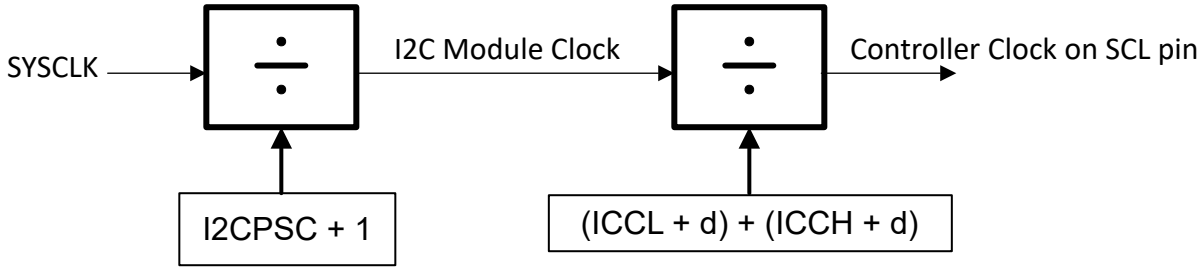


Figure 37-3. Clocking Diagram for the I2C Module

Note

To meet all of the I2C protocol timing specifications, the I2C module clock must be between 7 to 12MHz.

To specify the divide-down value, initialize the IPSC field of the prescaler register, I2CPSC. The resulting frequency is:

$$\text{I2C Module Clock (Fmod)} = \frac{\text{SYSCLK}}{(\text{I2CPSC} + 1)} \quad (35)$$

The prescaler must be initialized only while the I2C module is in the reset state (IRS = 0 in I2CMDR). The prescaled frequency takes effect only when IRS is changed to 1. Changing the IPSC value while IRS = 1 has no effect.

The controller clock appears on the SCL pin when the I2C module is configured to be a controller on the I2C bus. This clock controls the timing of communication between the I2C module and a target. As shown in Figure 37-3, a second clock divider in the I2C module divides down the module clock to produce the controller clock. The clock divider uses the ICCL value of I2CCLKL to divide down the low portion of the module clock signal and uses the ICCH value of I2CCLKH to divide down the high portion of the module clock signal. See Section 37.1.6 for the controller clock frequency equation.

37.1.6 I2C Clock Divider Registers (I2CCLKL and I2CCLKH)

As explained in Section 37.1.5, when the I2C module is a controller, the I2C module clock is divided down further to use as the controller clock on the SCL pin. As shown in Figure 37-4, the shape of the controller clock depends on two divide-down values:

- ICCL in I2CCLKL. For each controller clock cycle, ICCL determines the amount of time the signal is low.
- ICCH in I2CCKLH. For each controller clock cycle, ICCH determines the amount of time the signal is high.

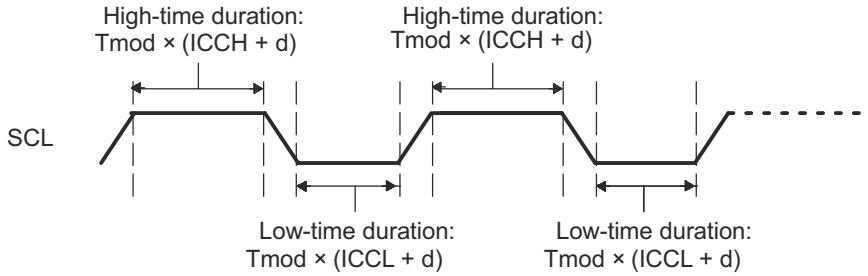


Figure 37-4. Roles of the Clock Divide-Down Values (ICCL and ICCH)

37.1.6.1 Formula for the Controller Clock Period

The controller clock period (Tmst) is a multiple of the period of the I2C Module Clock (Tmod):

$$\text{Controller Clock period (Tmst)} = \frac{[\,(\text{ICCH} + \text{d}) + (\text{ICCL} + \text{d})\,]}{\text{I2C Module Clock (Fmod)}} \quad (36)$$

where d depends on the divide-down value IPSC, as shown in Table 37-1. IPSC is described in the I2CPSC register.

Table 37-1. Dependency of Delay d on the Divide-Down Value IPSC

IPSC	d
0	7
1	6
Greater than 1	5

37.2 Configuring Device Pins

The GPIO mux registers must be configured to connect this peripheral to the device pins. To avoid glitches on the pins, the GPyGMUX bits must be configured first (while keeping the corresponding GPyMUX bits at the default of zero), followed by writing the GPyMUX register to the desired value.

Some IO functionality is defined by GPIO register settings independent of this peripheral. For input signals, the GPIO input qualification must be set to asynchronous mode by setting the appropriate GPxQSELn register bits to 11b. The internal pullups can be configured in the GPyPUD register.

See the *General-Purpose Input/Output (GPIO)* chapter for more details on GPIO mux and settings.

Note

To support a wider range of I2C IO levels, certain GPIOs have configurable fail-safe systems, V_{IH} minimum thresholds, and configurable sinking capabilities.

37.3 I2C Module Operational Details

This section provides an overview of the I2C bus protocol and how it is implemented.

37.3.1 Input and Output Voltage Levels

One clock pulse is generated by the controller device for each data bit transferred. Due to a variety of different technology devices that can be connected to the I2C bus, the levels of logic 0 (low) and logic 1 (high) are not fixed and depend on the associated level of V_{DD}. For details, see the device data sheet.

37.3.2 Selecting Pullup Resistors

The chosen pullup resistor must meet the I2C standard timings. In most circumstances, 2.2kΩ of total bus resistance to VDDIO is sufficient. The value of the pullup resistance used on both the SCL and SDA pins be matched is also recommended. For evaluating pullup resistor values for a particular design, see the *I2C Bus Pullup Resistor Calculation Application Report*.

37.3.3 Data Validity

The data on SDA must be stable during the high period of the clock (see Figure 37-5). The high or low state of the data line, SDA, must change only when the clock signal on SCL is low.

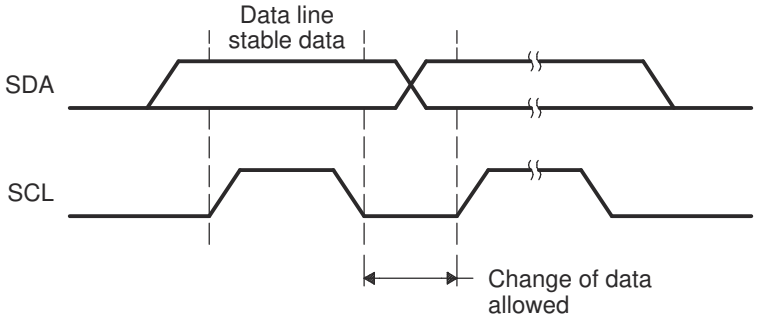


Figure 37-5. Bit Transfer on the I2C bus

37.3.4 Operating Modes

The I2C module has four basic operating modes to support data transfers as a controller and as a target. See Table 37-2 for the names and descriptions of the modes.

If the I2C module is a controller, the I2C module begins as a controller-transmitter and typically transmits an address for a particular target. When giving data to the target, the I2C module must remain a controller-transmitter. To receive data from a target, the I2C module must be changed to the controller-receiver mode.

If the I2C module is a target, the I2C module begins as a target-receiver and typically sends acknowledgment when the I2C module recognizes the target address from a controller. If the controller is sending data to the I2C module, the module must remain a target-receiver. If the controller has requested data from the I2C module, the module must be changed to the target-transmitter mode.

Table 37-2. Operating Modes of the I2C Module

Operating Mode	Description
Target-receiver mode	The I2C module is a target and receives data from a controller. All targets begin in this mode. In this mode, serial data bits received on SDA are shifted in with the clock pulses that are generated by the controller. As a target, the I2C module does not generate the clock signal, but can hold SCL low while the intervention of the device is required (RSFULL = 1 in I2CSTR) after a byte has been received. See Section 37.3.8 for more details.
Target-transmitter mode	The I2C module is a target and transmits data to a controller. This mode can be entered only from the target-receiver mode; the I2C module must first receive a command from the controller. When using any of the 7-bit/10-bit addressing formats, the I2C module enters the target-transmitter mode if the target address byte is the same as the address (in I2COAR) and the controller has transmitted R/ $\overline{W}$ = 1. As a target-transmitter, the I2C module then shifts the serial data out on SDA with the clock pulses that are generated by the controller. While a target, the I2C module does not generate the clock signal, but it can hold SCL low while the intervention of the device is required (XSMT = 0 in I2CSTR) after a byte has been transmitted. See Section 37.3.8 for more details.
Controller-receiver mode	The I2C module is a controller and receives data from a target. This mode can be entered only from the controller-transmitter mode; the I2C module must first transmit a command to the target. When using any of the 7-bit/10-bit addressing formats, the I2C module enters the controller-receiver mode after transmitting the target address byte and R/ $\overline{W}$ = 1. Serial data bits on SDA are shifted into the I2C module with the clock pulses generated by the I2C module on SCL. The clock pulses are inhibited and SCL is held low when the intervention of the device is required (RSFULL = 1 in I2CSTR) after a byte has been received.
Controller-transmitter mode	The I2C module is a controller and transmits control information and data to a target. All controllers begin in this mode. In this mode, data assembled in any of the 7-bit/10-bit addressing formats is shifted out on SDA. The bit shifting is synchronized with the clock pulses generated by the I2C module on SCL. The clock pulses are inhibited and SCL is held low when the intervention of the device is required (XSMT = 0 in I2CSTR) after a byte has been transmitted.

To summarize, SCL is held low in the following conditions:

- When an overrun condition is detected (RSFULL = 1), in Target-receiver mode.
- When an underflow condition is detected (XSMT = 0), in Target-transmitter mode.

I2C target nodes accept and provide data when the I2C controller node requests data.

- To release SCL in target-receiver mode, read data from I2CDRR.
- To release SCL in target-transmitter mode, write data to I2CDXR.
- To force a release without handling the data, reset the module using the I2CMDR.IRS bit.

Table 37-3. Controller-Transmitter/Receiver Bus Activity Defined by the RM, STT, and STP Bits of I2CMDR

RM	STT	STP	Bus Activity(1)	Description
0	0	0	None	No activity
0	0	1	P	STOP condition
0	1	0	S-A-D..(n)..D.	START condition, target address, n data bytes (n = value in I2CCNT)
0	1	1	S-A-D..(n)..D-P	START condition, target address, n data bytes, STOP condition (n = value in I2CCNT)
1	0	0	None	No activity
1	0	1	P	STOP condition
1	1	0	S-A-D-D-D.	Repeat mode transfer: START condition, target address, continuous data transfers until STOP condition or next START condition
1	1	1	None	Reserved bit combination (No activity)

(1) S = START condition; A = Address; D = Data byte; P = STOP condition;

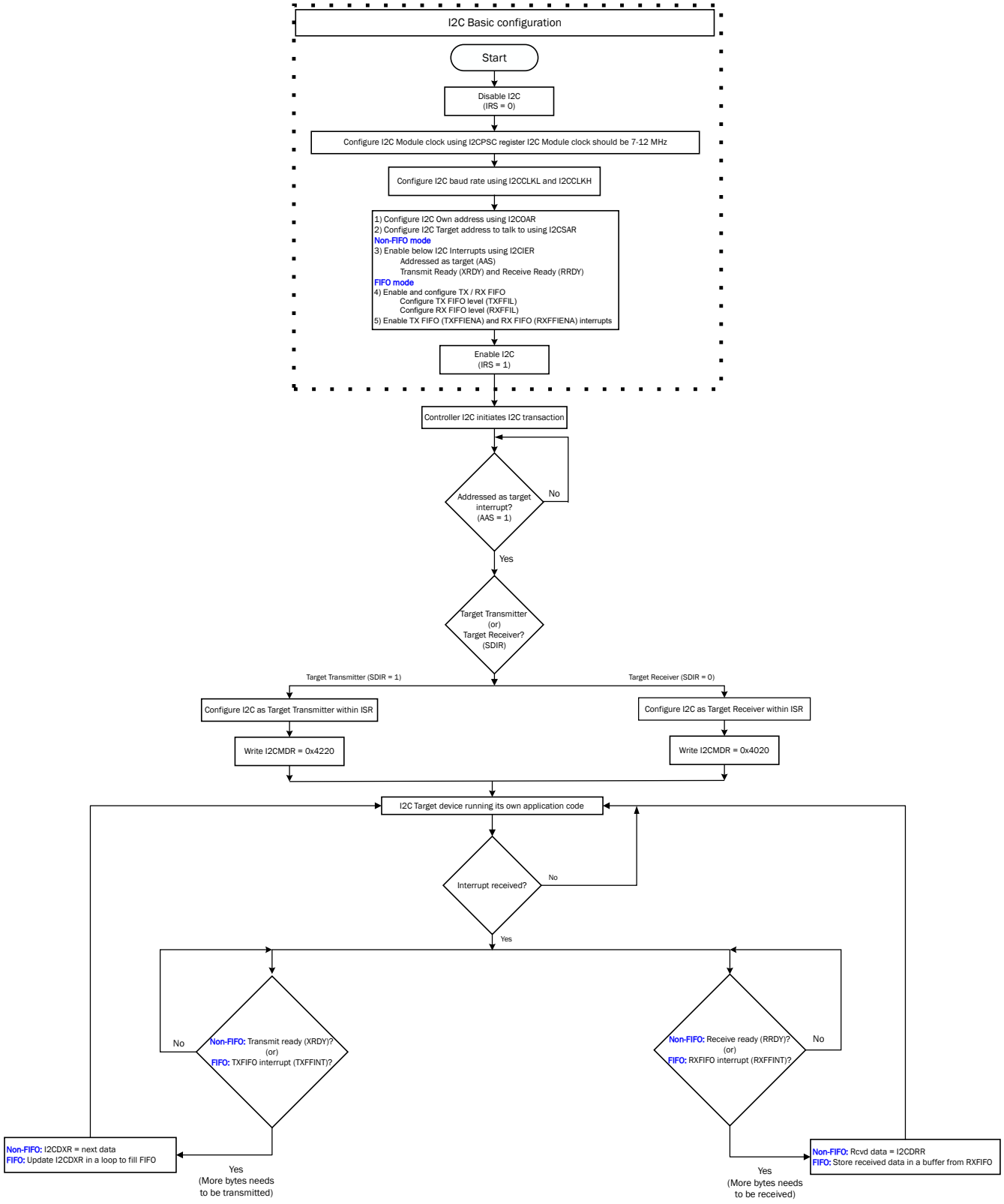


Figure 37-6. I2C Target TX / RX Flowchart

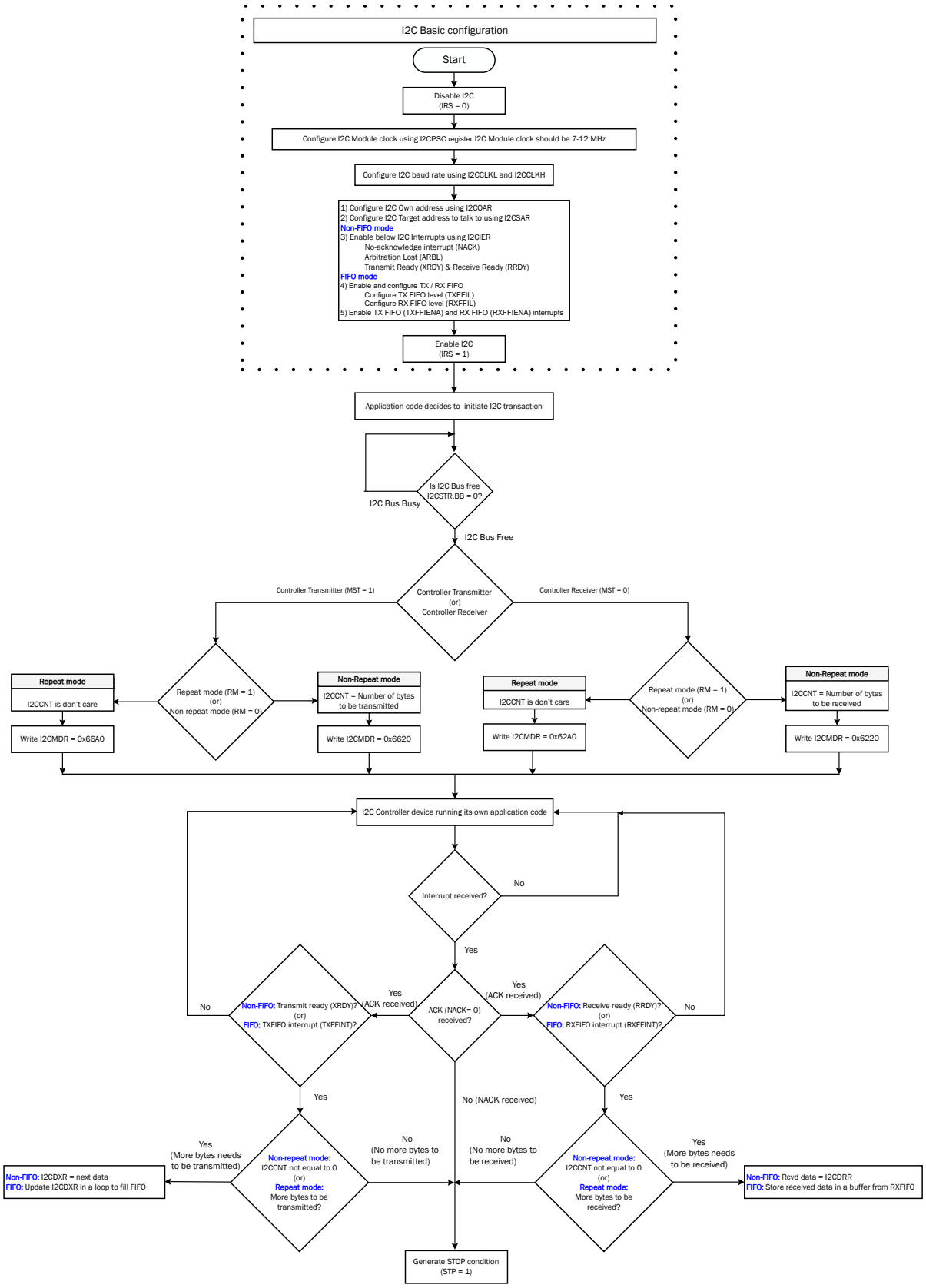


Figure 37-7. I2C Controller TX / RX Flowchart

37.3.5 I2C Module START and STOP Conditions

START and STOP conditions can be generated by the I2C module when the module is configured to be a controller on the I2C bus. As shown in Figure 37-8:

- The START condition is defined as a high-to-low transition on the SDA line while SCL is high. A controller drives this condition to indicate the start of a data transfer.
- The STOP condition is defined as a low-to-high transition on the SDA line while SCL is high. A controller drives this condition to indicate the end of a data transfer.

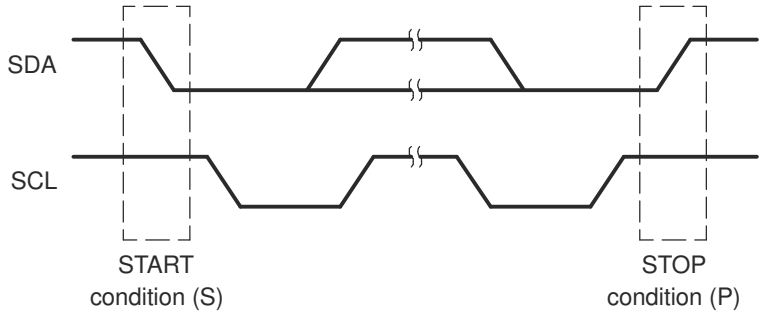


Figure 37-8. I2C Module START and STOP Conditions

After a START condition and before a subsequent STOP condition, the I2C bus is considered busy, and the bus busy (BB) bit of I2CSTR is 1. Between a STOP condition and the next START condition, the bus is considered free, and BB is 0.

For the I2C module to start a data transfer with a START condition, the controller mode bit (MST) and the START condition bit (STT) in I2CMDR must both be 1. For the I2C module to end a data transfer with a STOP condition, the STOP condition bit (STP) must be set to 1. When the BB bit is set to 1 and the STT bit is set to 1, a repeated START condition is generated. For a description of I2CMDR and the bits (including MST, STT, and STP), see Section 37.7.

The I2C peripheral cannot detect a START or STOP condition while in reset (IRS = 0). The BB bit remains in the cleared state (BB = 0) while the I2C peripheral is in reset (IRS = 0). When the I2C peripheral is taken out of reset (IRS set to 1), the BB bit does not correctly reflect the I2C bus status until a START or STOP condition is detected.

Follow these steps before initiating the first data transfer with I2C:

1. After taking the I2C peripheral out of reset by setting the IRS bit to 1, wait a period larger than the total time taken for the longest data transfer in the application. By waiting for a period of time after I2C comes out of reset, make sure that at least one START or STOP condition has occurred on the I2C bus and has been captured by the BB bit. After this period, the BB bit correctly reflects the state of the I2C bus.
2. Check the BB bit and verify that BB = 0 (bus not busy) before proceeding.
3. Begin data transfers.

Not resetting the I2C peripheral in between transfers makes sure that the BB bit reflects the actual bus status. If users must reset the I2C peripheral in between transfers, repeat steps 1 through 3 every time the I2C peripheral is taken out of reset.

37.3.6 Non-repeat Mode versus Repeat Mode

Non-repeat mode:

- When I2CMDR.RM = 0, I2C module is configured in non-repeat mode.
- I2CCNT register determines the number of bytes to be transmitted or received.
- If STP = 0 in I2CMDR, the ARDY bit is set when the internal data counter counts down to 0.
- If STP = 1, ARDY bit does not get set and I2C module generates a STOP condition when the internal data counter counts down to 0.

Note

In non-repeat mode (RM = 0), if I2CCNT is set to 0, I2C state machine expects to transmit or receive 65536 bytes and not 0 bytes.

Repeat mode:

- When I2CMDR.RM = 1, I2C module is configured in repeat mode.
- I2CCNT register contents do not determine the number of bytes to be transmitted or received.
- Number of bytes to be transmitted or received can be controlled by software.
- ARDY bit gets set at end of transmission and reception of each byte.

Note

Once you start I2C transaction in non-repeat mode or repeat mode, you cannot switch into another mode until the I2C transaction is completed with a STOP condition.

37.3.7 Serial Data Formats

Figure 37-9 shows an example of a data transfer on the I2C bus. The I2C module supports 1 to 8-bit data values. In Figure 37-9, 8-bit data is transferred. Each bit put on the SDA line equates to 1 pulse on the SCL line, and the values are always transferred with the most significant bit (MSB) first. The number of data values that can be transmitted or received is unrestricted. The serial data format used in Figure 37-9 is the 7-bit addressing format. The I2C module supports the formats shown in Figure 37-10 through Figure 37-12 and described in the paragraphs that follow the figures.

Note

In Figure 37-9 through Figure 37-12, n = the number of data bits (from 1 to 8) specified by the bit count (BC) field of I2CMDR.

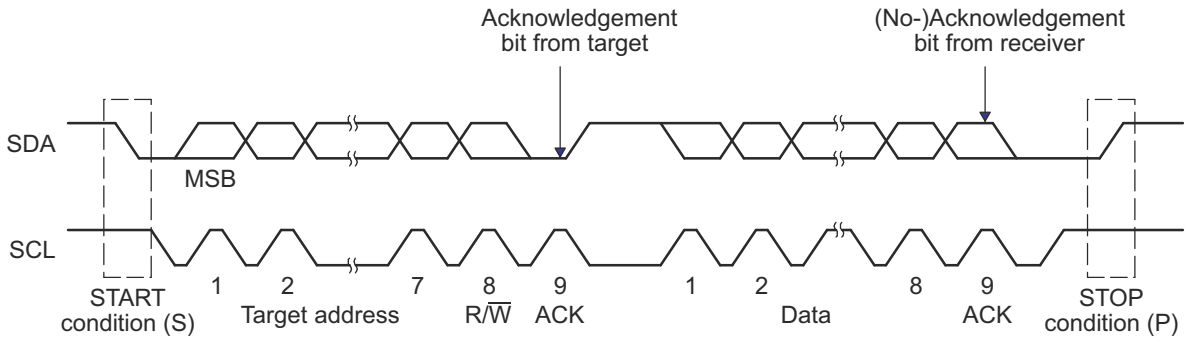


Figure 37-9. I2C Module Data Transfer (7-Bit Addressing with 8-bit Data Configuration Shown)

37.3.7.1 7-Bit Addressing Format

The 7-bit addressing format is the default format after reset. Disabling expanded address (I2CMDR.XA = 0) and free data format (I2CMDR.FDF = 0) enables 7-bit addressing format.

In this format (see Figure 37-10), the first byte after a START condition (S) consists of a 7-bit target address followed by a R/$\overline{W}$ bit. R/$\overline{W}$ determines the direction of the data:

- R/$\overline{W}$ = 0: The I2C controller writes (transmits) data to the addressed target. This can be achieved by setting I2CMDR.TRX = 1 (Transmitter mode)
- R/$\overline{W}$ = 1: The I2C controller reads (receives) data from the target. This can be achieved by setting I2CMDR.TRX = 0 (Receiver mode)

An extra clock cycle dedicated for acknowledgment (ACK) is inserted after each byte. If the ACK bit is inserted by the target after the first byte from the controller, it is followed by n bits of data from the transmitter (controller or target, depending on the R/$\overline{W}$ bit). n is a number from 1 to 8 determined by the bit count (BC) field of I2CMDR. After the data bits have been transferred, the receiver inserts an ACK bit.

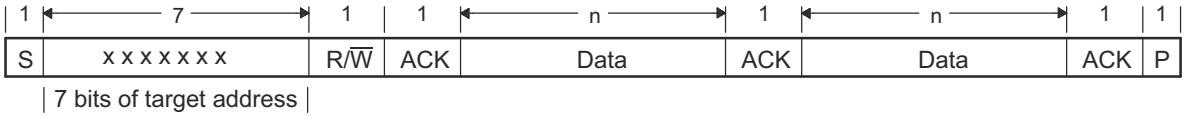


Figure 37-10. I2C Module 7-Bit Addressing Format (FDF = 0, XA = 0 in I2CMDR)

37.3.7.2 10-Bit Addressing Format

The 10-bit addressing format can be enabled by setting expanded address (I2CMDR.XA = 1) and disabling free data format (I2CMDR.FDF = 0).

The 10-bit addressing format (see Figure 37-11) is similar to the 7-bit addressing format, but the controller sends the target address in two separate byte transfers. The first byte consists of 11110b, the two MSBs of the 10-bit target address, and R/$\overline{W}$. The second byte is the remaining 8 bits of the 10-bit target address. The target must send acknowledgment after each of the two byte transfers. Once the controller has written the second byte to the target, the controller can either write data or use a repeated START condition to change the data direction. For more details about using 10-bit addressing, see the NXP Semiconductors I2C bus specification.

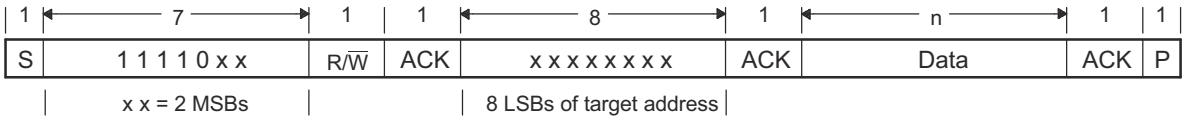


Figure 37-11. I2C Module 10-Bit Addressing Format (FDF = 0, XA = 1 in I2CMDR)

37.3.7.3 Free Data Format

The free data format can be enabled by setting I2CMDR. FDF = 1.

In this format (see Figure 37-12), the first byte after a START condition (S) is a data byte. An ACK bit is inserted after each data byte, which can be from 1 to 8 bits, depending on the BC field of I2CMDR. No address or data-direction bit is sent. Therefore, the transmitter and the receiver must both support the free data format, and the direction of the data must be constant throughout the transfer.

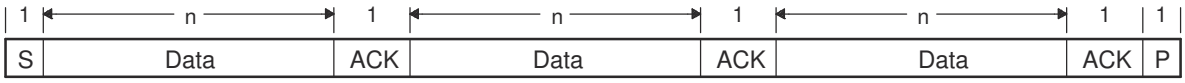


Figure 37-12. I2C Module Free Data Format (FDF = 1 in I2CMDR)

Note

The free data format is not supported in the digital loopback mode (I2CMDR.DLB = 1).

Table 37-4. How the MST and FDF Bits of I2CMDR Affect the Role of the TRX Bit of I2CMDR

MST	FDF	I2C Module State	Function of TRX
0	0	In target mode but not free data format mode	TRX is a don't care. Depending on the command from the controller, the I2C module responds as a receiver or a transmitter.
0	1	In target mode and free data format mode	The free data format mode requires that the I2C module remains the transmitter or the receiver throughout the transfer. TRX identifies the role of the I2C module: TRX = 1: The I2C module is a transmitter. TRX = 0: The I2C module is a receiver.
1	0	In controller mode but not free data format mode	TRX = 1: The I2C module is a transmitter. TRX = 0: The I2C module is a receiver.
1	1	In controller mode and free data format mode	TRX = 0: The I2C module is a receiver. TRX = 1: The I2C module is a transmitter.

37.3.7.4 Using a Repeated START Condition

I2C controller can communicate with multiple target addresses without having to give up control of the I2C bus by driving a STOP condition. This can be achieved by driving another START condition at the end of each data type. The repeated START condition can be used with the 7-bit addressing and 10-bit addressing. Figure 37-13 shows a repeated START condition in the 7-bit addressing format.

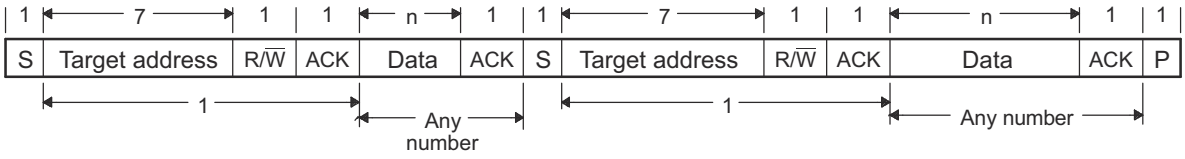


Figure 37-13. Repeated START Condition (in This Case, 7-Bit Addressing Format)

Note

In Figure 37-13, n = the number of data bits (from 1 to 8) specified by the bit count (BC) field of I2CMDR.

37.3.8 Clock Synchronization

Under normal conditions, only one controller device generates the clock signal, SCL. During the arbitration procedure, however, there are two or more controllers and the clock must be synchronized so that the data output can be compared. Figure 37-14 illustrates the clock synchronization. The wired-AND property of SCL means that a device that first generates a low period on SCL overrules the other devices. At this high-to-low transition, the clock generators of the other devices are forced to start a low period. The SCL is held low by the device with the longest low period. The other devices that finish the low periods must wait for SCL to be released, before starting the high periods. A synchronized signal on SCL is obtained, where the slowest device determines the length of the low period and the fastest device determines the length of the high period.

If a device pulls down the clock line for a longer time, the result is that all clock generators must enter the wait state. In this way, a target slows down a fast controller and the slow device creates enough time to store a received byte or to prepare a byte to be transmitted.

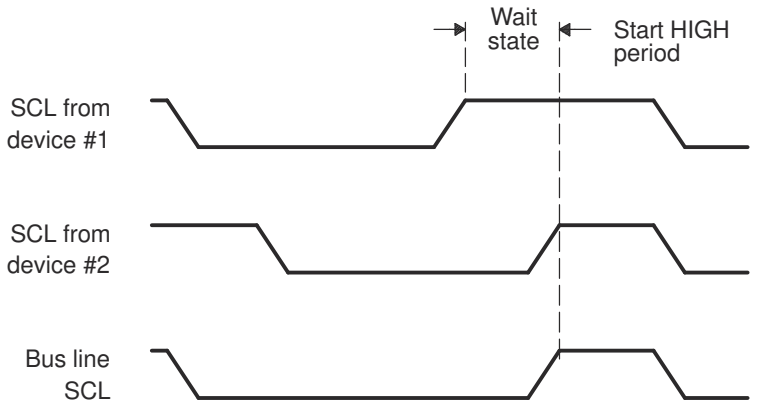


Figure 37-14. Synchronization of Two I2C Clock Generators During Arbitration

37.3.9 Clock Stretching

An I2C target device pulls the SCL line low to push the I2C controller device into the wait state preventing the I2C controller from transmitting data. This concept is called Clock Stretching.

The I2C supports three different types of clock stretching:

- **Automatic Clock Stretching:** Figure 37-15 shows the timing diagram for automatic clock stretching.
  - Target Receiver mode:
    - Non-FIFO mode: I2C clock stretches after receiving 2 bytes from the controller.
    - FIFO mode: I2C clock stretches after receiving 17 bytes from the controller.
  - Target Transmitter mode: I2C clock stretches, if I2C transmit buffer is empty

Note

Automatic Clock Stretching is enabled by default and cannot be disabled.

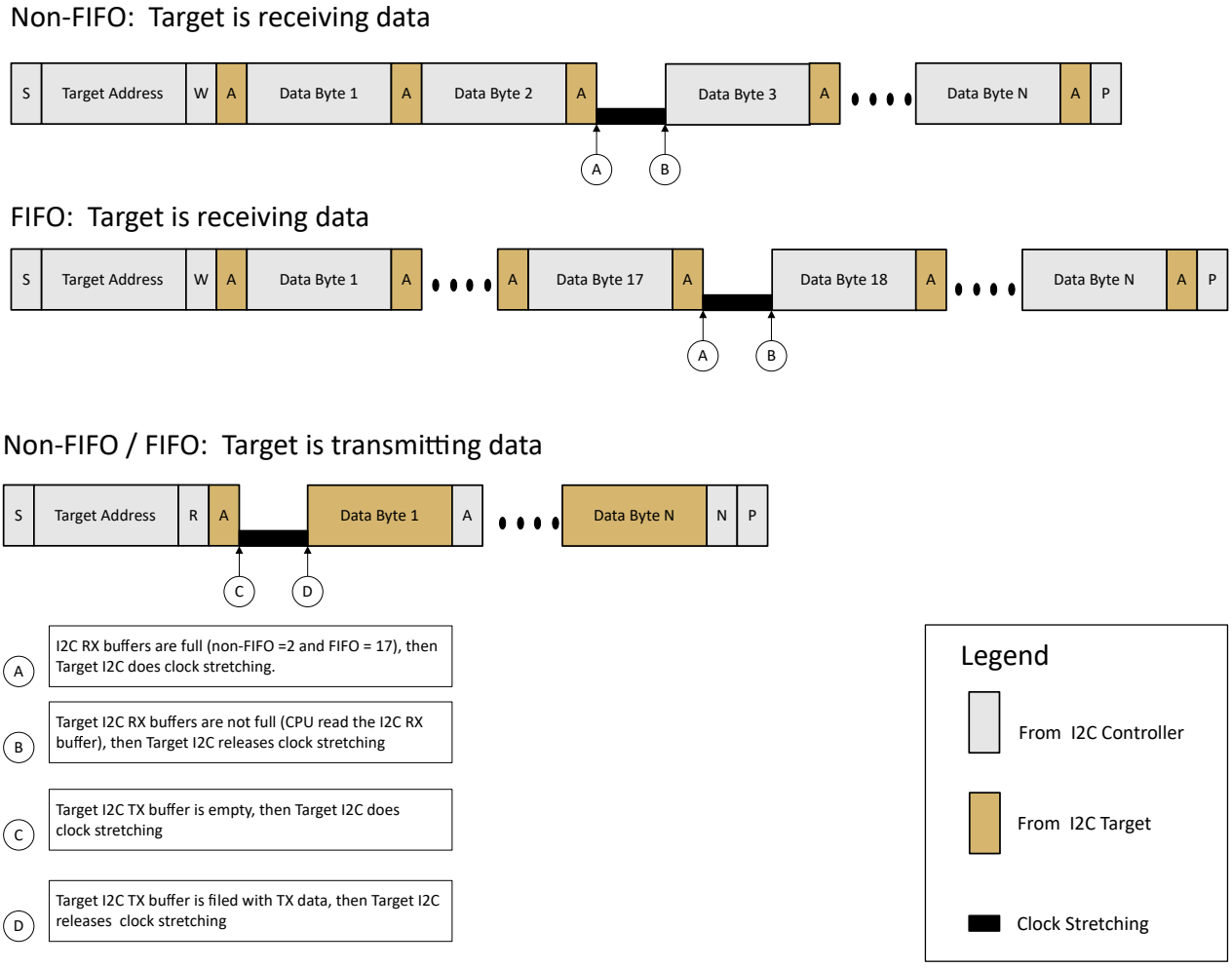


Figure 37-15. Automatic Clock Stretching

- **Extended Automatic Clock Stretching:** Figure 37-16 shows the timing diagram for extended automatic clock stretching.

  Extended Automatic Clock Stretching can be enabled by setting I2CEMDR. ECS = 1. Once this feature is enabled, I2C hardware automatically does clock stretching by pulling the SCL line low after every ACK/NACK cycle and generates an I2C interrupt. Inside the I2C ISR, I2CSTR.SCL_ECS is set to 1 to release the I2C from clock stretching.

Note

Extended Automatic Clock Stretching is disabled by default and can be enabled by setting I2CEMDR.ECS = 1.

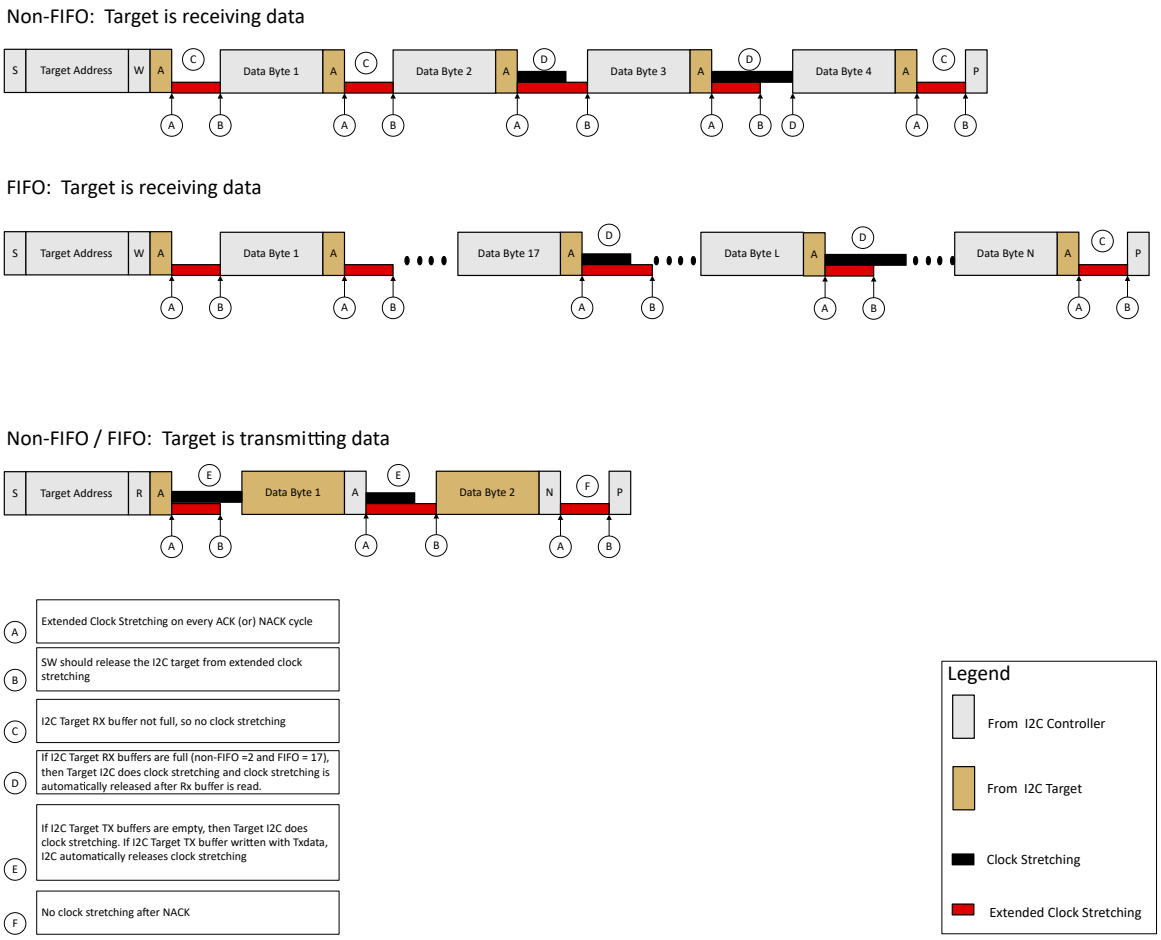


Figure 37-16. Extended Automatic Clock Stretching

- **Manual Clock Stretching:** Manual Clock Stretching can be enabled by setting I2CEMDR.MCS = 1 within an ISR routine. Once this feature is enabled, I2C hardware can be configured to manually do clock stretching by pulling the SCL line low after specific ACK/NACK cycles. I2C module then performs housekeeping and executes normal ISR sequence steps before disabling MCS by setting I2CEMDR.MCS = 0.

Note

Manual Clock Stretching is disabled by default and can be enabled by setting I2CEMDR.MCS = 1. Timing requirements for this type of clock stretching are not provided due to dependency on user configuration and application.

37.3.10 Arbitration

If two or more controller-transmitters attempt to start a transmission on the same bus at approximately the same time, an arbitration procedure is invoked. The arbitration procedure uses the data presented on the serial data bus (SDA) by the competing transmitters. Figure 37-17 illustrates the arbitration procedure between two devices. The first controller-transmitter that releases the SDA line high is overruled by another controller-transmitter that drives the SDA low. The arbitration procedure gives priority to the device that transmits the serial data stream with the lowest binary value. If two or more devices send identical first bytes, arbitration continues on the subsequent bytes.

If the I2C module is the losing controller, the I2c module switches to the target-receiver mode, sets the arbitration lost (ARBL) flag, and generates the arbitration-lost interrupt request.

If during a serial transfer the arbitration procedure is still in progress when a repeated START condition or a STOP condition is transmitted to SDA, the controller-transmitters involved must send the repeated START condition or the STOP condition at the same position in the format frame. Arbitration is not allowed between:

- A repeated START condition and a data bit
- A STOP condition and a data bit
- A repeated START condition and a STOP condition

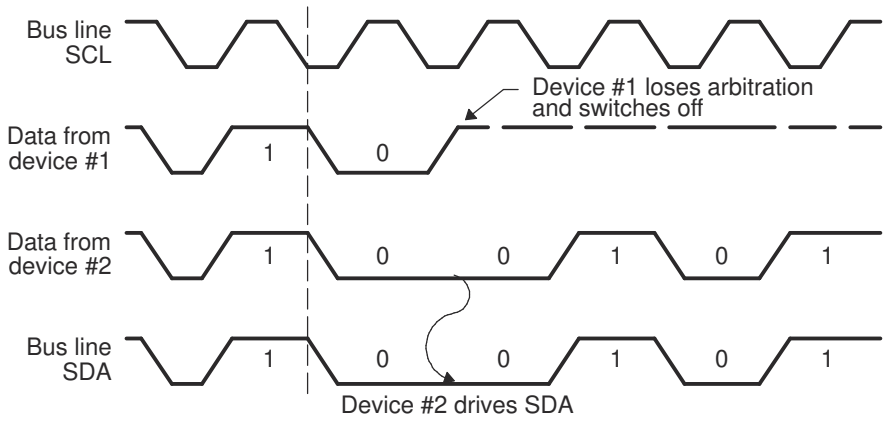


Figure 37-17. Arbitration Procedure Between Two Controller-Transmitters

37.3.11 Digital Loopback Mode

The I2C module support a self-test mode called digital loopback, which is enabled by setting the DLB bit in the I2CMDR register. In this mode, data transmitted out of the I2CDXR register is received in the I2CDRR register. The data follows an internal path, and takes n cycles to reach I2CDRR, where:

n = 8 * (SYSCLK) / (I2C module clock (Fmod))

The transmit clock and the receive clock are the same. The address seen on the external SDA pin is the address in the I2COAR register. Figure 37-18 shows the signal routing in digital loopback mode.

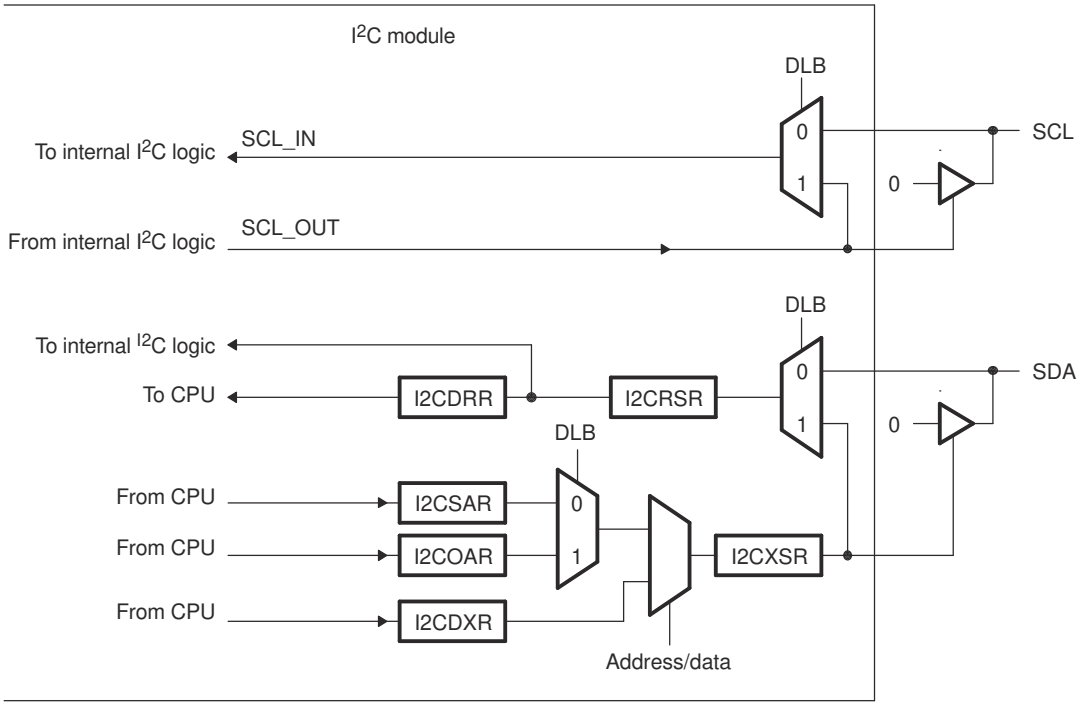


Figure 37-18. Pin Diagram Showing the Effects of the Digital Loopback Mode (DLB) Bit

Note

The free data format (I2CMDR.FDF = 1) is not supported in digital loopback mode.

37.3.12 NACK Bit Generation

When the I2C module is a receiver (controller or target), the I2C module can acknowledge or ignore bits sent by the transmitter. To ignore any new bits, the I2C module must send a no-acknowledge (NACK) bit during the acknowledge cycle on the bus. Table 37-5 summarizes the various ways you can allow the I2C module to send a NACK bit.

Note

When a NACK is sent, the following occurs:

1. The STP in I2CMDR is cleared
2. SCL is held low
3. The NACK in I2CSTR is set

Table 37-5. Ways to Generate a NACK Bit

I2C Module Condition	NACK Bit Generation Options
Target-receiver modes	Allow an overrun condition (RSFULL = 1 in I2CSTR) Reset the module (IRS = 0 in I2CMDR) Set the NACKMOD bit of I2CMDR before the rising edge of the last data bit you intend to receive
Controller-receiver mode AND Repeat mode (RM = 1 in I2CMDR)	Generate a STOP condition (STP = 1 in I2CMDR) Reset the module (IRS = 0 in I2CMDR) Set the NACKMOD bit of I2CMDR before the rising edge of the last data bit you intend to receive
Controller-receiver mode AND Nonrepeat mode (RM = 0 in I2CMDR)	If STP = 1 in I2CMDR, allow the internal data counter to count down to 0 and thus force a STOP condition If STP = 0, make STP = 1 to generate a STOP condition Reset the module (IRS = 0 in I2CMDR) Set STP = 1 to generate a STOP condition Set the NACKMOD bit of I2CMDR before the rising edge of the last data bit you intend to receive

37.4 Interrupt Requests Generated by the I2C Module

Each I2C module can generate two CPU interrupts.

1. Basic I2C interrupt: Possible basic I2C interrupt sources that can trigger this interrupt are described in Section 37.4.1.
2. I2C FIFO interrupt: Possible I2C FIFO interrupt sources that can trigger this interrupt are described in Section 37.4.2

37.4.1 Basic I2C Interrupt Requests

The I2C module generates the interrupt requests described in Table 37-6. As shown in Figure 37-19, all requests are multiplexed through an arbiter to a single I2C interrupt request to the CPU. Each interrupt request has a flag bit in the status register (I2CSTR) and an enable bit in the interrupt enable register (I2CIER). When one of the specified events occurs, the flag bit is set. If the corresponding enable bit is 0, the interrupt request is blocked. If the enable bit is 1, the request is forwarded to the CPU as an I2C interrupt.

The I2C interrupt is one of the maskable interrupts of the CPU. As with any maskable interrupt request, if the request is properly enabled in the CPU, the CPU executes the corresponding interrupt service routine (I2CINT1A_ISR). The I2CINT1A_ISR for the I2C interrupt can determine the interrupt source by reading the interrupt source register, I2CISRC. Then the I2CINT1A_ISR can branch to the appropriate subroutine.

After the CPU reads I2CISRC, the following events occur:

1. The flag for the source interrupt is cleared in I2CSTR. Exception: The ARDY, RRDY, and XRDY bits in I2CSTR are not cleared when I2CISRC is read. To clear one of these bits, write a 1 to the bit.
2. The arbiter determines which of the remaining interrupt requests has the highest priority, writes the code for that interrupt to I2CISRC, and forwards the interrupt request to the CPU.

Table 37-6. Descriptions of the Basic I2C Interrupt Requests

I2C Interrupt Request	Interrupt Source
XRDYINT	Transmit ready condition: The data transmit register (I2CDXR) is ready to accept new data because the previous data has been copied from I2CDXR to the transmit shift register (I2CXSR). As an alternative to using XRDYINT, the CPU can poll the XRDY bit of the status register, I2CSTR. XRDYINT must not be used when in FIFO mode. Use the FIFO interrupts instead.
RRDYINT	Receive ready condition: The data receive register (I2CDRR) is ready to be read because data has been copied from the receive shift register (I2CRSR) to I2CDRR. As an alternative to using RRDYINT, the CPU can poll the RRDY bit of I2CSTR. RRDYINT must not be used when in FIFO mode. Use the FIFO interrupts instead.
ARDYINT	Register-access ready condition: The I2C module registers are ready to be accessed because the previously programmed address, data, and command values have been used. The specific events that generate ARDYINT are the same events that set the ARDY bit of I2CSTR. As an alternative to using ARDYINT, the CPU can poll the ARDY bit.
NACKINT	No-acknowledgment condition: The I2C module is configured as a controller-transmitter and did not received acknowledgment from the target-receiver. As an alternative to using NACKINT, the CPU can poll the NACK bit of I2CSTR.
ARBLINT	Arbitration-lost condition: The I2C module has lost an arbitration contest with another controller-transmitter. As an alternative to using ARBLINT, the CPU can poll the ARBL bit of I2CSTR.
SCDINT	Stop condition detected: A STOP condition was detected on the I2C bus. As an alternative to using SCDINT, the CPU can poll the SCD bit of the status register, I2CSTR.
AASINT	Addressed as target condition: The I2C has been addressed as a target device by another controller on the I2C bus. As an alternative to using AASINT, the CPU can poll the AAS bit of the status register, I2CSTR.
SCLECSINT	SCL Extended Automatic Clock Stretching: SCL line is pulled low in extended automatic clock stretching mode. As an alternative to using SCLECSINT, the CPU can poll the SCL_ECS bit of the status register, I2CSTR.

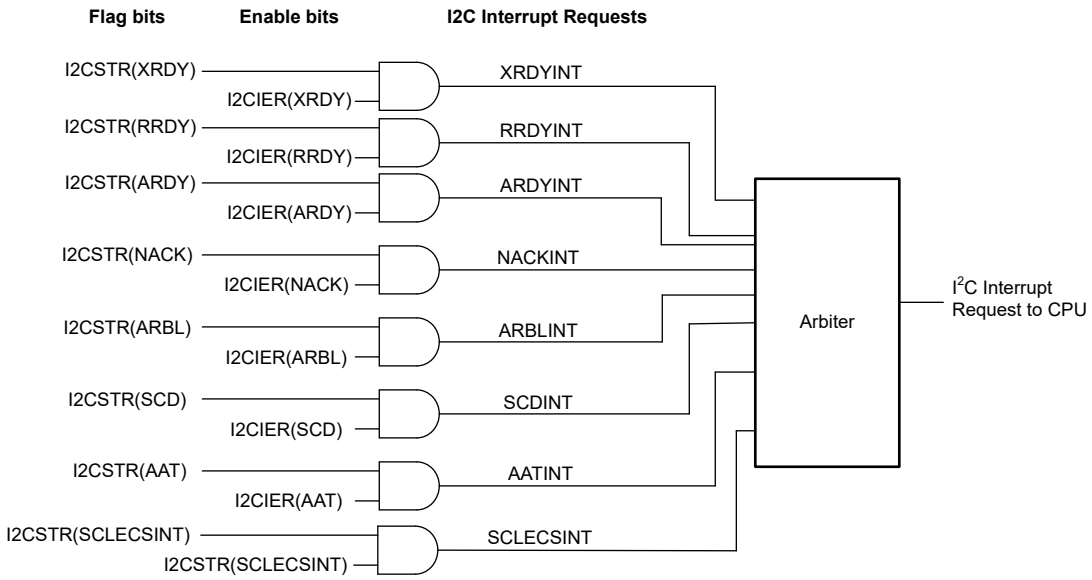


Figure 37-19. Enable Paths of the I2C Interrupt Requests

The priorities of the basic I2C interrupt requests are listed in order of highest priority to lowest priority:

1. ARBLINT
2. NACKINT
3. ARDYINT
4. RRDYINT
5. XRDYINT
6. SCDINT
7. AATINT
8. SCLECSINT

The I2C module has a backwards compatibility bit (BC) in the I2CEMDR register. The timing diagram in demonstrates the effect the backwards compatibility bit has on I2C module registers and interrupts when configured as a target-transmitter.

37.4.2 I2C FIFO Interrupts

In addition to the seven basic I2C interrupts, the transmit and receive FIFOs each contain the ability to generate an interrupt (I2CINT2A). The transmit FIFO can be configured to generate an interrupt after transmitting a defined number of bytes, up to 16. The receive FIFO can be configured to generate an interrupt after receiving a defined number of bytes, up to 16. These two interrupt sources are ORed together into a single maskable CPU interrupt. Figure 37-20 shows the structure of I2C FIFO interrupt. The interrupt service routine can then read the FIFO interrupt status flags to determine from which source the interrupt came. See the I2C transmit FIFO register (I2CFFTX) and the I2C receive FIFO register (I2CFFRX) descriptions.

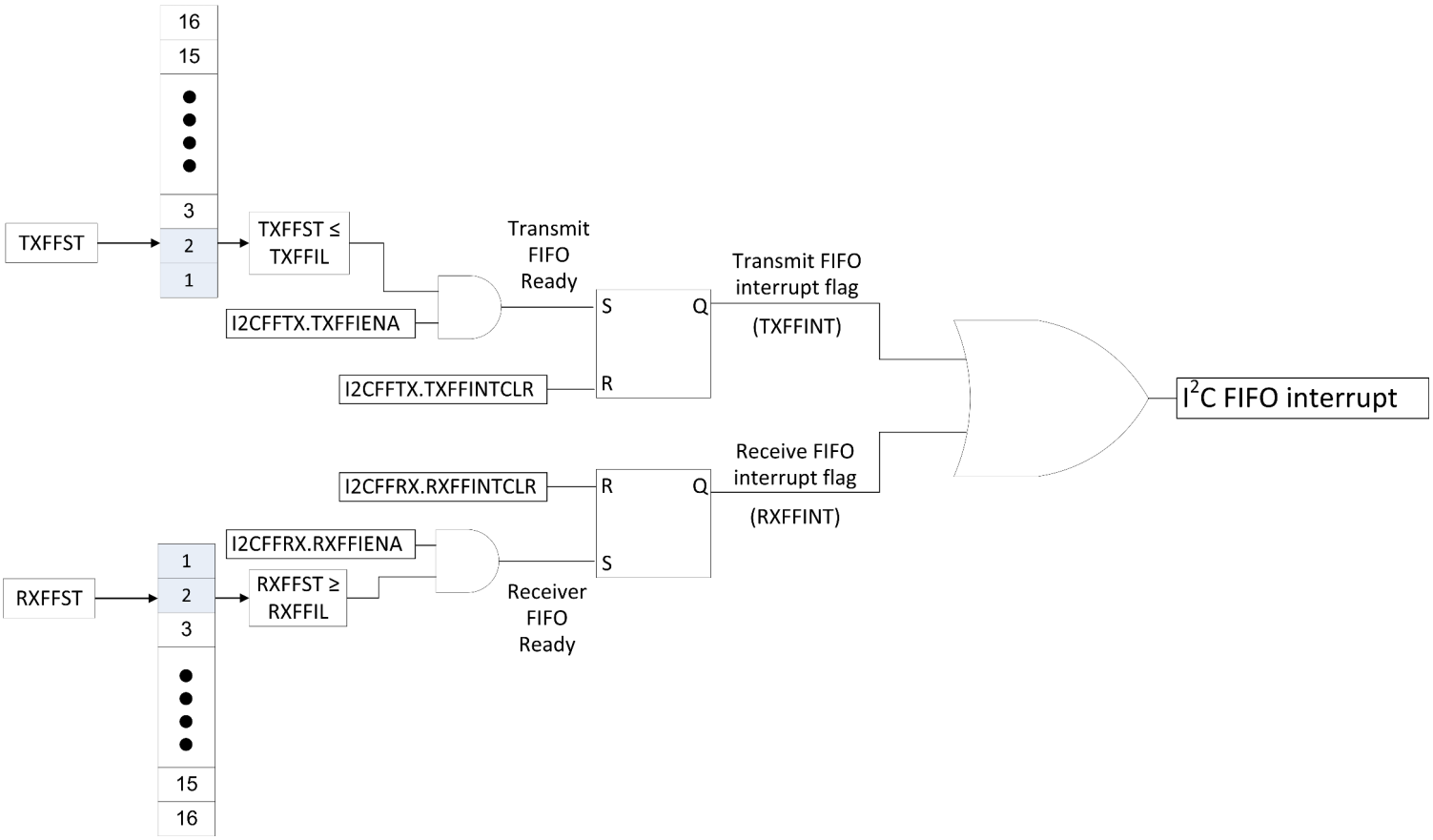


Figure 37-20. I2C FIFO Interrupt

37.5 Resetting or Disabling the I2C Module

You can reset or disable the I2C module in two ways:

- Write 0 to the I2C reset bit (IRS) in the I2C mode register (I2CMDR). All status bits (in I2CSTR) are forced to the default values, and the I2C module remains disabled until IRS is changed to 1. The SDA and SCL pins are in the high-impedance state.
- Initiate a device reset by driving the $\overline{\text{XRS}}$ pin low. The entire device is reset and is held in the reset state until you drive the pin high. When the $\overline{\text{XRS}}$ pin is released, all I2C module registers are reset to the default values. The IRS bit is forced to 0, which resets the I2C module. The I2C module stays in the reset state until you write 1 to IRS.

The IRS must be 0 while you configure or reconfigure the I2C module. Forcing IRS to 0 can be used to save power and to clear error conditions.

37.6 Software

37.6.1 I2C Registers to Driverlib Functions

Table 37-7. I2C Registers to Driverlib Functions

File	Driverlib Function
OAR	
i2c.h	I2C_setOwnAddress
IER	
i2c.c	I2C_enableInterrupt
i2c.c	I2C_disableInterrupt
STR	
i2c.c	I2C_getInterruptStatus
i2c.c	I2C_clearInterruptStatus
i2c.h	I2C_isBusBusy
i2c.h	I2C_getStatus
i2c.h	I2C_clearStatus
CLKL	
i2c.c	I2C_initController
i2c.c	I2C_initControllerModuleFrequency
CLKH	
i2c.c	I2C_initController
i2c.c	I2C_initControllerModuleFrequency
CNT	
i2c.h	I2C_setDataCount
DRR	
i2c.h	I2C_getData
TAR	
i2c.h	I2C_setTargetAddress
DXR	
i2c.h	I2C_putData
MDR	
i2c.h	I2C_enableModule
i2c.h	I2C_disableModule
i2c.h	I2C_setConfig
i2c.h	I2C_setBitCount
i2c.h	I2C_sendStartCondition
i2c.h	I2C_sendStopCondition
i2c.h	I2C_sendNACK
i2c.h	I2C_getStopConditionStatus
i2c.h	I2C_setAddressMode
i2c.h	I2C_setEmulationMode
i2c.h	I2C_enableLoopback
i2c.h	I2C_disableLoopback
ISRC	
i2c.h	I2C_getInterruptSource
EMDR	
i2c.h	I2C_setExtendedMode

Table 37-7. I2C Registers to Driverlib Functions (continued)

File	Driverlib Function
i2c.h	I2C_enableExtendedAutomaticClkStretchingMode
i2c.h	I2C_disableExtendedAutomaticClkStretchingMode
i2c.h	I2C_enableManualClkStretchingMode
i2c.h	I2C_disableManualClkStretchingMode
i2c.h	I2C_enableNACKCompatibilityMode
i2c.h	I2C_disableNACKCompatibilityMode
PSC	
i2c.c	I2C_initController
i2c.c	I2C_initControllerModuleFrequency
i2c.c	I2C_configureModuleFrequency
i2c.c	I2C_configureModuleClockFrequency
i2c.h	I2C_getPreScaler
FFTX	
i2c.c	I2C_enableInterrupt
i2c.c	I2C_disableInterrupt
i2c.c	I2C_getInterruptStatus
i2c.c	I2C_clearInterruptStatus
i2c.h	I2C_enableFIFO
i2c.h	I2C_disableFIFO
i2c.h	I2C_setFIFOInterruptLevel
i2c.h	I2C_getFIFOInterruptLevel
i2c.h	I2C_getTxFIFOStatus
FFRX	
i2c.c	I2C_enableInterrupt
i2c.c	I2C_disableInterrupt
i2c.c	I2C_getInterruptStatus
i2c.c	I2C_clearInterruptStatus
i2c.h	I2C_enableFIFO
i2c.h	I2C_disableFIFO
i2c.h	I2C_setFIFOInterruptLevel
i2c.h	I2C_getFIFOInterruptLevel
i2c.h	I2C_getRxFIFOStatus

37.6.2 I2C Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/i2c

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

37.6.2.1 I2C Digital Loopback with FIFO Interrupts - SINGLE_CORE

FILE: i2c_ex1_loopback.c

This program uses the internal loopback test mode of the I2C module. Both the TX and RX I2C FIFOs and their interrupts are used. The pinmux and I2C initialization is done through the sysconfig file.

A stream of data is sent and then compared to the received stream. The sent data looks like this:
0000 0001
0001 0002
0002 0003
....
00FE 00FF
00FF 0000
etc..
This pattern is repeated forever.

External Connections
- None

Watch Variables
- *sData* - Data to send
- *rData* - Received data
- *rDataPoint* - Used to keep track of the last position in the receive stream for error checking

37.6.2.2 I2C EEPROM - SINGLE_CORE

FILE: i2c_ex2_eeprom.c

This program will write 1-14 words to EEPROM and read them back. The data written and the EEPROM address written to are contained in the message structure, i2cMsgOut. The data read back will be contained in the message structure i2cMsgIn.

External Connections
- Connect external I2C EEPROM at address 0x50
- Connect DEVICE_GPIO_PIN_SDAA on to external EEPROM SDA (serial data) pin
- Connect DEVICE_GPIO_PIN_SCLA on to external EEPROM SCL (serial clock) pin

Watch Variables
- *i2cMsgOut* - Message containing data to write to EEPROM
- *i2cMsgIn* - Message containing data read from EEPROM

37.6.2.3 I2C Digital External Loopback with FIFO Interrupts - SINGLE_CORE

FILE: i2c_ex3_external_loopback.c

This program uses the I2CA and I2CB modules for achieving external loopback. The I2CA TX FIFO and the I2CB RX FIFO are used along with their interrupts.

A stream of data is sent on I2CA and then compared to the received stream on I2CB. The sent data looks like this:
0000 0001
0001 0002
0002 0003
....

00FE 00FF
00FF 0000
etc..
This pattern is repeated forever.

External Connections

- Connect SCLA(DEVICE_GPIO_PIN_SCLA) to SCLB (DEVICE_GPIO_PIN_SCLB)
- and SDAA(DEVICE_GPIO_PIN_SDAA) to SDAB (DEVICE_GPIO_PIN_SDAB)
- Connect DEVICE_GPIO_PIN_LED1 to an LED used to depict data transfers.

Watch Variables

- *sData* - Data to send
- *rData* - Received data
- *rDataPoint* - Used to keep track of the last position in the receive stream for error checking

37.6.2.4 I2C Extended Clock Stretching Controller TX - SINGLE_CORE

FILE: i2c_ex7_clock_stretching_controller_tx.c

This program uses the extended clock stretching mode of the I2C module. Both the TX and RX I2C Non-FIFOs and their interrupts are used.

A stream of data is sent and then compared to the received stream.

37.6.2.5 I2C Extended Clock Stretching Target RX - SINGLE_CORE

FILE: i2c_ex7_clock_stretching_target_rx.c

This program uses the extended clock stretching mode of the I2C module. Both the TX and RX I2C Non-FIFOs and their interrupts are used.

A stream of data is sent and then compared to the received stream.

37.7 I2C Registers

This Section describes the I2C Registers.

37.7.1 I2C Base Address Table

Table 37-8. I2C Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
I2C_REGS	I2CA_BASE	0x7015_0000	YES	YES	YES	YES	YES	YES	-	YES
I2C_REGS	I2CB_BASE	0x7015_1000	YES	YES	YES	YES	YES	YES	-	YES

37.7.2 I2C_REGS Registers

Table 37-9 lists the memory-mapped registers for the I2C_REGS registers. All register offset addresses not listed in Table 37-9 should be considered as reserved locations and the register contents should not be modified.

Table 37-9. I2C_REGS Registers

Offset	Acronym	Register Name	Protection
0h	I2COAR	I2C Own address	
2h	I2CIER	I2C Interrupt Enable	
4h	I2CSTR	I2C Status	
6h	I2CCLKL	I2C Clock low-time divider	
8h	I2CCLKH	I2C Clock high-time divider	
Ah	I2CCNT	I2C Data count	
Ch	I2CDRR	I2C Data receive	
Eh	I2CTAR	I2C TARGET address	
10h	I2CDXR	I2C Data Transmit	
12h	I2CMDR	I2C Mode	
14h	I2CISRC	I2C Interrupt Source	
16h	I2CEMDR	I2C Extended Mode	
18h	I2CPSC	I2C Prescaler	
40h	I2CFFTX	I2C FIFO Transmit	
42h	I2CFFRX	I2C FIFO Receive	

Complex bit access types are encoded to fit into small table cells. Table 37-10 shows the codes that are used for access types in this section.

Table 37-10. I2C_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

37.7.2.1 I2COAR Register (Offset = 0h) [Reset = 0000h]

I2COAR is shown in Figure 37-21 and described in Table 37-11.

Return to the Summary Table.

The I2C own address register (I2COAR) is a 16-bit register. The I2C module uses this register to specify its own TARGET address, which distinguishes it from other TARGETs connected to the I2C-bus. If the 7-bit addressing mode is selected (XA = 0 in I2CMDR), only bits 6-0 are used
write 0s to bits 9-7.

Figure 37-21. I2COAR Register

15	14	13	12	11	10	9	8
RESERVED						OAR	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
OAR							
R/W-0h							

Table 37-11. I2COAR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	OAR	R/W	0h	In 7-bit addressing mode (XA = 0 in I2CMDR): 00h-7Fh Bits 6-0 provide the 7-bit TARGET address of the I2C module. Write 0s to bits 9-7. In 10-bit addressing mode (XA = 1 in I2CMDR): 000h-3FFh Bits 9-0 provide the 10-bit TARGET address of the I2C module. Reset type: SYSRSn

37.7.2.2 I2CIER Register (Offset = 2h) [Reset = 0000h]

I2CIER is shown in Figure 37-22 and described in Table 37-12.

Return to the Summary Table.

I2CIER is used by the CPU to individually enable or disable I2C interrupt requests.

Figure 37-22. I2CIER Register

15	14	13	12	11	10	9	8
SCL_ECS	RESERVED						
R/W-0h	R-0h						
7	6	5	4	3	2	1	0
RESERVED	AAT	SCD	XRDY	RRDY	ARDY	NACK	ARBL
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 37-12. I2CIER Register Field Descriptions

Bit	Field	Type	Reset	Description
15	SCL_ECS	R/W	0h	SCL Extended Automatic Clock Stretch interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt request disabled 1h (R/W) = Interrupt request enabled
14-7	RESERVED	R	0h	Reserved
6	AAT	R/W	0h	Addressed as TARGET interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt request disabled 1h (R/W) = Interrupt request enabled
5	SCD	R/W	0h	Stop condition detected interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt request disabled 1h (R/W) = Interrupt request enabled
4	XRDY	R/W	0h	Transmit-data-ready interrupt enable bit. This bit should not be set when using FIFO mode. Reset type: SYSRSn 0h (R/W) = Interrupt request disabled 1h (R/W) = Interrupt request enabled
3	RRDY	R/W	0h	Receive-data-ready interrupt enable bit. This bit should not be set when using FIFO mode. Reset type: SYSRSn 0h (R/W) = Interrupt request disabled 1h (R/W) = Interrupt request enabled
2	ARDY	R/W	0h	Register-access-ready interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt request disabled 1h (R/W) = Interrupt request enabled
1	NACK	R/W	0h	No-acknowledgment interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt request disabled 1h (R/W) = Interrupt request enabled
0	ARBL	R/W	0h	Arbitration-lost interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt request disabled 1h (R/W) = Interrupt request enabled

37.7.2.3 I2CSTR Register (Offset = 4h) [Reset = 0410h]

I2CSTR is shown in Figure 37-23 and described in Table 37-13.

Return to the Summary Table.

The I2C status register (I2CSTR) is a 16-bit register used to determine which interrupt has occurred and to read status information.

Figure 37-23. I2CSTR Register

15	14	13	12	11	10	9	8
SCL_ECS	TDIR	NACKSNT	BB	RSFULL	XSMT	AAT	AD0
R/W1C-0h	R/W1C-0h	R/W1C-0h	R-0h	R-0h	R-1h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	BYTESENT	SCD	XRDY	RRDY	ARDY	NACK	ARBL
R-0h	R/W1C-0h	R/W1C-0h	R-1h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

Table 37-13. I2CSTR Register Field Descriptions

Bit	Field	Type	Reset	Description
15	SCL_ECS	R/W1C	0h	SCL Status bit in extended automatic clock stretching mode 0: SCL line is pulled high. 1: SCL line is pulled low after ACK (or) NACK phase. Writing '1' to this bit releases SCL. SCL line is pulled high. Note: This bit is applicable only for extended clock stretching mode Reset type: SYSRSn 0h (R/W) = SCL line is not pulled low in extended automatic clock stretching mode. SCL_ECS bit is cleared by one o fthe following events: - It is manually cleared. To clear this bit, write a 1 to it. - The I2C module is reset. 1h (R/W) = SCL line is pulled low in extended automatic clock stretching mode.
14	TDIR	R/W1C	0h	TARGET direction bit Reset type: SYSRSn 0h (R/W) = I2C is not addressed as a TARGET transmitter. TDIR is cleared by one of the following events: - It is manually cleared. To clear this bit, write a 1 to it. - Digital loopback mode is enabled. - A START or STOP condition occurs on the I2C bus. 1h (R/W) = I2C is addressed as a TARGET transmitter.
13	NACKSNT	R/W1C	0h	NACK sent bit. This bit is used when the I2C module is in the receiver mode. One instance in which NACKSNT is affected is when the NACK mode is used (see the description for NACKMOD in Reset type: SYSRSn 0h (R/W) = NACK not sent. NACKSNT bit is cleared by any one of the following events: - It is manually cleared. To clear this bit, write a 1 to it. - The I2C module is reset (either when 0 is written to the IRS bit of I2CMDR or when the whole device is reset). 1h (R/W) = NACK sent: A no-acknowledge bit was sent during the acknowledge cycle on the I2C-bus.

Table 37-13. I2CSTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	BB	R	0h	Bus busy bit. BB indicates whether the I2C-bus is busy or is free for another data transfer. See the paragraph following the table for more information Reset type: SYSRSn 0h (R/W) = Bus free. BB is cleared by any one of the following events: - The I2C module receives or transmits a STOP bit (bus free). - The I2C module is reset. 1h (R/W) = Bus busy: The I2C module has received or transmitted a START bit on the bus.
11	RSFULL	R	0h	Receive shift register full bit. RSFULL indicates an overrun condition during reception. Overrun occurs when new data is received into the shift register (I2CRSR) and the old data has not been read from the receive register (I2CDRR). As new bits arrive from the SDA pin, they overwrite the bits in I2CRSR. The new data will not be copied to ICDRR until the previous data is read. Reset type: SYSRSn 0h (R/W) = No overrun detected. RSFULL is cleared by any one of the following events: - I2CDRR is read by the CPU. Emulator reads of the I2CDRR do not affect this bit. - The I2C module is reset. 1h (R/W) = Overrun detected
10	XSMT	R	1h	Transmit shift register empty bit. XSMT = 0 indicates that the transmitter has experienced underflow. Underflow occurs when the transmit shift register (I2CXSR) is empty but the data transmit register (I2CDXR) has not been loaded since the last I2CDXR-to-I2CXSR transfer. The next I2CDXR-to-I2CXSR transfer will not occur until new data is in I2CDXR. If new data is not transferred in time, the previous data may be re-transmitted on the SDA pin. Reset type: SYSRSn 0h (R/W) = Underflow detected (empty) 1h (R/W) = No underflow detected (not empty). XSMT is set by one of the following events: - Data is written to I2CDXR. - The I2C module is reset
9	AAT	R	0h	Addressed-as-TARGET bit Reset type: SYSRSn 0h (R/W) = In the 7-bit addressing mode, the AAT bit is cleared when receiving a NACK, a STOP condition, or a repeated START condition. In the 10-bit addressing mode, the AAT bit is cleared when receiving a NACK, a STOP condition, or by a TARGET address different from the I2C peripheral's own TARGET address. 1h (R/W) = The I2C module has recognized its own TARGET address or an address of all zeros (general call).
8	AD0	R	0h	Address 0 bits Reset type: SYSRSn 0h (R/W) = AD0 has been cleared by a START or STOP condition. 1h (R/W) = An address of all zeros (general call) is detected.
7	RESERVED	R	0h	Reserved

Table 37-13. I2CSTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	BYTESENT	R/W1C	0h	Byte Transmit over indication. BYTESENT is set when the CONTROLLER/TARGET has successfully sent the byte on SCL/SDA lines. This is diagnostic register which needs to be explicitly cleared by Software. In case not cleared the stale status would keep reflecting as no automated clear incorporated to avoid corner conditions. Reset type: SYSRSn 0h (R/W) = The I2C module has not finished transmitting the next data byte. BYTESENT is cleared by any one of the following events: - It is manually cleared. To clear this bit, write a 1 to it. - The I2C module is reset. 1h (R/W) = The I2C module has completed the transmission of a byte.
5	SCD	R/W1C	0h	Stop condition detected bit. SCD is set when the I2C sends or receives a STOP condition. The I2C module delays clearing of the I2CMDR[STP] bit until the SCD bit is set. Reset type: SYSRSn 0h (R/W) = STOP condition not detected since SCD was last cleared. SCD is cleared by any one of the following events: - I2CISRC is read by the CPU when it contains the value 110b (stop condition detected). Emulator reads of the I2CISRC do not affect this bit. - SCD is manually cleared. To clear this bit, write a 1 to it. - The I2C module is reset. 1h (R/W) = A STOP condition has been detected on the I2C bus.
4	XRDY	R	1h	Transmit-data-ready interrupt flag bit. When not in FIFO mode, XRDY indicates that the data transmit register (I2CDXR) is ready to accept new data. FCM=0 : When the previous data has been copied from I2CDXR to the transmit shift register (I2CXSR). The CPU can poll XRDY or use the XRDY interrupt request When in FIFO mode, use TXFFINT instead. FCM=1: XRDY is asserted only when next data is required it gets deasserted with write to I2CDXR. Both Polling and interrupt based data transfers are allowed in the FCM mode. Reset type: SYSRSn 0h (R/W) = I2CDXR not ready. XRDY is cleared when data is written to I2CDXR. 1h (R/W) = I2CDXR ready: Data has been copied from I2CDXR to I2CXSR. XRDY is also forced to 1 when the I2C module is reset.
3	RRDY	R/W1C	0h	Receive-data-ready interrupt flag bit. When not in FIFO mode, RRDY indicates that the data receive register (I2CDRR) is ready to be read because data has been copied from the receive shift register (I2CRSR) to I2CDRR. The CPU can poll RRDY or use the RRDY interrupt request When in FIFO mode, use RXFFINT instead. Reset type: SYSRSn 0h (R/W) = I2CDRR not ready. RRDY is cleared by any one of the following events: - I2CDRR is read by the CPU. Emulator reads of the I2CDRR do not affect this bit. - RRDY is manually cleared. To clear this bit, write a 1 to it. - The I2C module is reset. 1h (R/W) = I2CDRR ready: Data has been copied from I2CRSR to I2CDRR.

Table 37-13. I2CSTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	ARDY	R/W1C	0h	Register-access-ready interrupt flag bit (only Applicable when the I2C module is in the CONTROLLER mode). ARDY indicates that the I2C module registers are ready to be accessed because the previously programmed address, data, and command values have been used. The CPU can poll ARDY or use the ARDY interrupt request Reset type: SYSRSn 0h (R/W) = The registers are not ready to be accessed. ARDY is cleared by any one of the following events: - The I2C module starts using the current register contents. - ARDY is manually cleared. To clear this bit, write a 1 to it. - The I2C module is reset. 1h (R/W) = The registers are ready to be accessed. In the nonrepeat mode (RM = 0 in I2CMDR): If STP = 0 in I2CMDR, the ARDY bit is set when the internal data counter counts down to 0. If STP = 1, ARDY is not affected (instead, the I2C module generates a STOP condition when the counter reaches 0). In the repeat mode (RM = 1): ARDY is set at the end of each byte transmitted from I2CDXR.
1	NACK	R/W1C	0h	No-acknowledgment interrupt flag bit. NACK applies when the I2C module is a CONTROLLER transmitter (or) TARGET transmitter. NACK indicates whether the I2C module has detected an acknowledge bit (ACK) or a noacknowledge bit (NACK). The CPU can poll NACK or use the NACK interrupt request. Reset type: SYSRSn 0h (R/W) = ACK received/NACK not received. This bit is cleared by any one of the following events: - An acknowledge bit (ACK) has been sent by the TARGET receiver. - NACK is manually cleared. To clear this bit, write a 1 to it. - The CPU reads the interrupt source register (I2CISRC) and the register contains the code for a NACK interrupt. Emulator reads of the I2CISRC do not affect this bit. - The I2C module is reset. 1h (R/W) = NACK bit received. The hardware detects that a no-acknowledge (NACK) bit has been received. Note: While the I2C module performs a general call transfer, NACK is 1, even if one or more TARGETs send acknowledgment.
0	ARBL	R/W1C	0h	Arbitration-lost interrupt flag bit (only applicable when the I2C module is a CONTROLLER-transmitter). ARBL primarily indicates when the I2C module has lost an arbitration contest with another CONTROLLERtransmitter. The CPU can poll ARBL or use the ARBL interrupt request. Reset type: SYSRSn 0h (R/W) = Arbitration not lost. AL is cleared by any one of the following events: - AL is manually cleared. To clear this bit, write a 1 to it. - The CPU reads the interrupt source register (I2CISRC) and the register contains the code for an AL interrupt. Emulator reads of the I2CISRC do not affect this bit. - The I2C module is reset. 1h (R/W) = Arbitration lost. AL is set by any one of the following events: - The I2C module senses that it has lost an arbitration with two or more competing transmitters that started a transmission almost simultaneously. - The I2C module attempts to start a transfer while the BB (bus busy) bit is set to 1. When AL becomes 1, the CNT and STP bits of I2CMDR are cleared, and the I2C module becomes a TARGET-receiver.

37.7.2.4 I2CCLKL Register (Offset = 6h) [Reset = 0000h]

I2CCLKL is shown in Figure 37-24 and described in Table 37-14.

Return to the Summary Table.

I2C Clock low-time divider

Figure 37-24. I2CCLKL Register

15	14	13	12	11	10	9	8
I2CCLKL							
R/W-0h							
7	6	5	4	3	2	1	0
I2CCLKL							
R/W-0h							

Table 37-14. I2CCLKL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	I2CCLKL	R/W	0h	Clock low-time divide-down value. To produce the low time duration of the CONTROLLER clock, the period of the module clock is multiplied by (ICCL + d). d is an adjustment factor based on the prescaler. See the Clock Divider Registers section of the Introduction for details. Note: These bits must be set to a non-zero value for proper I2C clock generation. Reset type: SYSRSn

37.7.2.5 I2CCLKH Register (Offset = 8h) [Reset = 0000h]

I2CCLKH is shown in Figure 37-25 and described in Table 37-15.

Return to the Summary Table.

I2C Clock high-time divider

Figure 37-25. I2CCLKH Register

15	14	13	12	11	10	9	8
I2CCLKH							
R/W-0h							
7	6	5	4	3	2	1	0
I2CCLKH							
R/W-0h							

Table 37-15. I2CCLKH Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	I2CCLKH	R/W	0h	Clock high-time divide-down value. To produce the high time duration of the CONTROLLER clock, the period of the module clock is multiplied by (ICCL + d). d is an adjustment factor based on the prescaler. See the Clock Divider Registers section of the Introduction for details. Note: These bits must be set to a non-zero value for proper I2C clock generation. Reset type: SYSRSn

37.7.2.6 I2CCNT Register (Offset = Ah) [Reset = 0000h]

I2CCNT is shown in Figure 37-26 and described in Table 37-16.

Return to the Summary Table.

I2CCNT is a 16-bit register used to indicate how many data bytes to transfer when the I2C module is configured as a transmitter, or to receive when configured as a CONTROLLER receiver. In the repeat mode (RM = 1), I2CCNT is not used.
The value written to I2CCNT is copied to an internal data counter. The internal data counter is decremented by 1 for each byte transferred (I2CCNT remains unchanged). If a STOP condition is requested in the CONTROLLER mode (STP = 1 in I2CMDR), the I2C module terminates the transfer with a STOP condition when the countdown is complete (that is, when the last byte has been transferred).

Figure 37-26. I2CCNT Register

15	14	13	12	11	10	9	8
I2CCNT							
R/W-0h							
7	6	5	4	3	2	1	0
I2CCNT							
R/W-0h							

Table 37-16. I2CCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	I2CCNT	R/W	0h	Data count value. I2CCNT indicates the number of data bytes to transfer or receive. If a STOP condition is specified (STP=1) then I2CCNT will decrease after each byte is sent until it reaches zero, which in turn will generate a STOP condition. The value in I2CCNT is a don't care when the RM bit in I2CMDR is set to 1. Reset type: SYSRSn 0h (R/W) = data count value is 65536 1h (R/W) = data count value is 1 2h (R/W) = data count value is 2 FFFFh (R/W) = data count value is 65535

37.7.2.7 I2CDRR Register (Offset = Ch) [Reset = 0000h]

I2CDRR is shown in Figure 37-27 and described in Table 37-17.

Return to the Summary Table.

I2CDRR is a 16-bit register used by the CPU to read received data. The I2C module can receive a data byte with 1 to 8 bits. The number of bits is selected with the bit count (BC) bits in I2CMDR. One bit at a time is shifted in from the SDA pin to the receive shift register (I2CRSR). When a complete data byte has been received, the I2C module copies the data byte from I2CRSR to I2CDRR. The CPU cannot access I2CRSR directly.
If a data byte with fewer than 8 bits is in I2CDRR, the data value is right-justified, and the other bits of I2CDRR(7-0) are undefined. For example, if BC = 011 (3-bit data size), the receive data is in I2CDRR(2-0), and the content of I2CDRR(7-3) is undefined.
When in the receive FIFO mode, the I2CDRR register acts as the receive FIFO buffer.

Figure 37-27. I2CDRR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
DATA							
R-0h							

Table 37-17. I2CDRR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	DATA	R	0h	Receive data Reset type: SYSRSn

37.7.2.8 I2CTAR Register (Offset = Eh) [Reset = 03FFh]

I2CTAR is shown in Figure 37-28 and described in Table 37-18.

Return to the Summary Table.

The I2C TARGET address register (I2CSAR) is a 16-bit register for storing the next TARGET address that will be transmitted by the I2C module when it is a CONTROLLER. The SAR field of I2CSAR contains a 7-bit or 10-bit TARGET address. When the I2C module is not using the free data format (FDF = 0 in I2CMDR), it uses this address to initiate data transfers with a TARGET, or TARGETs. When the address is nonzero, the address is for a particular TARGET. When the address is 0, the address is a general call to all TARGETs. If the 7-bit addressing mode is selected (XA = 0 in I2CMDR), only bits 6-0 of I2CSAR are used
write 0s to bits 9-7.

Figure 37-28. I2CTAR Register

15	14	13	12	11	10	9	8
RESERVED						TAR	
R-0h						R/W-3FFh	
7	6	5	4	3	2	1	0
TAR							
R/W-3FFh							

Table 37-18. I2CTAR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	TAR	R/W	3FFh	In 7-bit addressing mode (XA = 0 in I2CMDR): 00h-7Fh Bits 6-0 provide the 7-bit TARGET address that the I2C module transmits when it is in the CONTROLLER-transmitter mode. Write 0s to bits 9-7. In 10-bit addressing mode (XA = 1 in I2CMDR): 000h-3FFh Bits 9-0 provide the 10-bit TARGET address that the I2C module transmits when it is in the CONTROLLER transmitter mode. Reset type: SYSRSn

37.7.2.9 I2CDXR Register (Offset = 10h) [Reset = 0000h]

I2CDXR is shown in Figure 37-29 and described in Table 37-19.

Return to the Summary Table.

The CPU writes transmit data to I2CDXR. This 16-bit register accepts a data byte with 1 to 8 bits. Before writing to I2CDXR, specify how many bits are in a data byte by loading the appropriate value into the bit count (BC) bits of I2CMDR. When writing a data byte with fewer than 8 bits, make sure the value is right-aligned in I2CDXR. After a data byte is written to I2CDXR, the I2C module copies the data byte to the transmit shift register (I2CXSR). The CPU cannot access I2CXSR directly. From I2CXSR, the I2C module shifts the data byte out on the SDA pin, one bit at a time.
When in the transmit FIFO mode, the I2CDXR register acts as the transmit FIFO buffer.

Figure 37-29. I2CDXR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
DATA							
R/W-0h							

Table 37-19. I2CDXR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	DATA	R/W	0h	Transmit data Reset type: SYSRSn

37.7.2.10 I2CMDR Register (Offset = 12h) [Reset = 0000h]

I2CMDR is shown in Figure 37-30 and described in Table 37-20.

Return to the Summary Table.

The I2C mode register (I2CMDR) is a 16-bit register that contains the control bits of the I2C module.

Figure 37-30. I2CMDR Register

15	14	13	12	11	10	9	8
NACKMOD	FREE	STT	RESERVED	STP	CNT	TRX	XA
R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RM	DLB	IRS	STB	FDF	BC		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		

Table 37-20. I2CMDR Register Field Descriptions

Bit	Field	Type	Reset	Description
15	NACKMOD	R/W	0h	NACK mode bit. This bit is only applicable when the I2C module is acting as a receiver. Reset type: SYSRSn 0h (R/W) = In the TARGET-receiver mode: The I2C module sends an acknowledge (ACK) bit to the transmitter during each acknowledge cycle on the bus. The I2C module only sends a no-acknowledge (NACK) bit if you set the NACKMOD bit. In the CONTROLLER-receiver mode: The I2C module sends an ACK bit during each acknowledge cycle until the internal data counter counts down to 0. At that point, the I2C module sends a NACK bit to the transmitter. To have a NACK bit sent earlier, you must set the NACKMOD bit 1h (R/W) = In either TARGET-receiver or CONTROLLER-receiver mode: The I2C module sends a NACK bit to the transmitter during the next acknowledge cycle on the bus. Once the NACK bit has been sent, NACKMOD is cleared. Important: To send a NACK bit in the next acknowledge cycle, you must set NACKMOD before the rising edge of the last data bit.
14	FREE	R/W	0h	This bit controls the action taken by the I2C module when a debugger breakpoint is encountered. Reset type: SYSRSn 0h (R/W) = When I2C module is CONTROLLER: If SCL is low when the breakpoint occurs, the I2C module stops immediately and keeps driving SCL low, whether the I2C module is the transmitter or the receiver. If SCL is high, the I2C module waits until SCL becomes low and then stops. When I2C module is TARGET: A breakpoint forces the I2C module to stop when the current transmission/reception is complete. 1h (R/W) = The I2C module runs free that is, it continues to operate when a breakpoint occurs.
13	STT	R/W	0h	START condition bit (only applicable when the I2C module is a CONTROLLER). The RM, STT, and STP bits determine when the I2C module starts and stops data transmissions (see Table 9-6). Note that the STT and STP bits can be used to terminate the repeat mode, and that this bit is not writable when IRS = 0. Reset type: SYSRSn 0h (R/W) = In the CONTROLLER mode, STT is automatically cleared after the START condition has been generated. 1h (R/W) = In the CONTROLLER mode, setting STT to 1 causes the I2C module to generate a START condition on the I2C-bus
12	RESERVED	R	0h	Reserved

Table 37-20. I2CMDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	STP	R/W	0h	STOP condition bit (only applicable when the I2C module is a CONTROLLER). In the CONTROLLER mode, the RM,STT, and STP bits determine when the I2C module starts and stops data transmissions. Note that the STT and STP bits can be used to terminate the repeat mode, and that this bit is not writable when IRS=0. When in non-repeat mode, at least one byte must be transferred before a stop condition can be generated. The I2C module delays clearing of this bit until after the I2CSTR[SCD] bit is set. To avoid disrupting the I2C state machine, the user must wait until this bit is clear before initiating a new message. Reset type: SYSRSn 0h (R/W) = STP is automatically cleared after the STOP condition has been generated 1h (R/W) = STP has been set by the device to generate a STOP condition when the internal data counter of the I2C module counts down to 0.
10	CNT	R/W	0h	CONTROLLER mode bit. CNT determines whether the I2C module is in the TARGET mode or the CONTROLLER mode. CNT is automatically changed from 1 to 0 when the I2C CONTROLLER generates a STOP condition Reset type: SYSRSn 0h (R/W) = TARGET mode. The I2C module is a TARGET and receives the serial clock from the CONTROLLER. 1h (R/W) = CONTROLLER mode. The I2C module is a CONTROLLER and generates the serial clock on the SCL pin.
9	TRX	R/W	0h	Transmitter mode bit. When relevant, TRX selects whether the I2C module is in the transmitter mode or the receiver mode. Reset type: SYSRSn 0h (R/W) = Receiver mode. The I2C module is a receiver and receives data on the SDA pin. 1h (R/W) = Transmitter mode. The I2C module is a transmitter and transmits data on the SDA pin.
8	XA	R/W	0h	Expanded address enable bit. Reset type: SYSRSn 0h (R/W) = 7-bit addressing mode (normal address mode). The I2C module transmits 7-bit TARGET addresses (from bits 6-0 of I2CTAR), and its own TARGET address has 7 bits (bits 6-0 of I2COAR). 1h (R/W) = 10-bit addressing mode (expanded address mode). The I2C module transmits 10-bit TARGET addresses (from bits 9-0 of I2CTAR), and its own TARGET address has 10 bits (bits 9-0 of I2COAR).
7	RM	R/W	0h	Repeat mode bit (only applicable when the I2C module is a CONTROLLER-transmitter or CONTROLLER-receiver). The RM, STT, and STP bits determine when the I2C module starts and stops data transmissions Reset type: SYSRSn 0h (R/W) = Nonrepeat mode. The value in the data count register (I2CCNT) determines how many bytes are received/transmitted by the I2C module. 1h (R/W) = Repeat mode. A data byte is transmitted each time the I2CDXR register is written to (or until the transmit FIFO is empty when in FIFO mode) until the STP bit is manually set. The value of I2CCNT is ignored. The ARDY bit/interrupt can be used to determine when the I2CDXR (or FIFO) is ready for more data, or when the data has all been sent and the CPU is allowed to write to the STP bit.

Table 37-20. I2CMDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	DLB	R/W	0h	Digital loopback mode bit. Reset type: SYSRSn 0h (R/W) = Digital loopback mode is disabled. 1h (R/W) = Digital loopback mode is enabled. For proper operation in this mode, the CNT bit must be 1. In the digital loopback mode, data transmitted out of I2CDXR is received in I2CDRR after n device cycles by an internal path, where: n = ((I2C input clock frequency/module clock frequency) x 8) The transmit clock is also the receive clock. The address transmitted on the SDA pin is the address in I2COAR. Note: The free data format (FDF = 1) is not supported in the digital loopback mode.
5	IRS	R/W	0h	I2C module reset bit. Reset type: SYSRSn 0h (R/W) = The I2C module is in reset/disabled. When this bit is cleared to 0, all status bits (in I2CSTR) are set to their default values. 1h (R/W) = The I2C module is enabled. This has the effect of releasing the I2C bus if the I2C peripheral is holding it.
4	STB	R/W	0h	START byte mode bit. This bit is only applicable when the I2C module is a CONTROLLER. As described in version 2.1 of the Philips Semiconductors I2C-bus specification, the START byte can be used to help a TARGET that needs extra time to detect a START condition. When the I2C module is a TARGET, it ignores a START byte from a CONTROLLER, regardless of the value of the STB bit. Reset type: SYSRSn 0h (R/W) = The I2C module is not in the START byte mode. 1h (R/W) = The I2C module is in the START byte mode. When you set the START condition bit (STT), the I2C module begins the transfer with more than just a START condition. Specifically, it generates: 1. A START condition 2. A START byte (0000 0001b) 3. A dummy acknowledge clock pulse 4. A repeated START condition Then, as normal, the I2C module sends the TARGET address that is in I2CTAR.
3	FDF	R/W	0h	Free data format mode bit. Reset type: SYSRSn 0h (R/W) = Free data format mode is disabled. Transfers use the 7-/10-bit addressing format selected by the XA bit. 1h (R/W) = Free data format mode is enabled. Transfers have the free data (no address) format described in Section 9.2.5. The free data format is not supported in the digital loopback mode (DLB=1).

Table 37-20. I2CMDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	BC	R/W	0h	Bit count bits. BC defines the number of bits (1 to 8) in the next data byte that is to be received or transmitted by the I2C module. The number of bits selected with BC must match the data size of the other device. Notice that when BC = 000b, a data byte has 8 bits. BC does not affect address bytes, which always have 8 bits. Note: If the bit count is less than 8, receive data is right-justified in I2CDRR(7-0), and the other bits of I2CDRR(7-0) are undefined. Also, transmit data written to I2CDXR must be right-justified Reset type: SYSRSn 0h (R/W) = 8 bits per data byte 1h (R/W) = 1 bit per data byte 2h (R/W) = 2 bits per data byte 3h (R/W) = 3 bits per data byte 4h (R/W) = 4 bits per data byte 5h (R/W) = 5 bits per data byte 6h (R/W) = 6 bits per data byte 7h (R/W) = 7 bits per data byte

37.7.2.11 I2CISRC Register (Offset = 14h) [Reset = 0000h]

I2CISRC is shown in Figure 37-31 and described in Table 37-21.

Return to the Summary Table.

The I2C interrupt source register (I2CISRC) is a 16-bit register used by the CPU to determine which event generated the I2C interrupt.

Figure 37-31. I2CISRC Register

15	14	13	12	11	10	9	8
RESERVED				WRITE_ZEROS			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED				INTCODE			
R-0h				R-0h			

Table 37-21. I2CISRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-8	WRITE_ZEROS	R/W	0h	TI internal testing bits These reserved bit locations should always be written as zeros. Reset type: SYSRSn
7-4	RESERVED	R	0h	Reserved
3-0	INTCODE	R	0h	Interrupt code bits. The binary code in INTCODE indicates the event that generated an I2C interrupt. A CPU read will clear this field. If another lower priority interrupt is pending and enabled, the value corresponding to that interrupt will then be loaded. Otherwise, the value will stay cleared. The interrupt events below are listed in descending order of priority. That is INTCODE 1 (Arbitration lost) has the highest priority and INTCODE 7 (Addressed as TARGET) has the lowest priority. In the case of an arbitration lost, a no-acknowledgment condition detected, or a stop condition detected, a CPU read will also clear the associated interrupt flag bit in the I2CSTR register. Emulator reads will not affect the state of this field or of the status bits in the I2CSTR register. Reset type: SYSRSn 0h (R/W) = None 1h (R/W) = Arbitration lost 2h (R/W) = No-acknowledgment condition detected 3h (R/W) = Registers ready to be accessed 4h (R/W) = Receive data ready 5h (R/W) = Transmit data ready 6h (R/W) = Stop condition detected 7h (R/W) = Addressed as TARGET 8h (R/W) = Extended Automatic Clock Stretching

37.7.2.12 I2CEMDR Register (Offset = 16h) [Reset = 0001h]

I2CEMDR is shown in Figure 37-32 and described in Table 37-22.

Return to the Summary Table.

I2C Extended Mode

Figure 37-32. I2CEMDR Register

15	14	13	12	11	10	9	8
RESERVED	RESERVED						NACK_CM
R/W-0h	R-0h						R/W-0h
7	6	5	4	3	2	1	0
SCLKEY				MCS	ECS	FCM	BC
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-1h

Table 37-22. I2CEMDR Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R/W	0h	Reserved
14-9	RESERVED	R	0h	Reserved
8	NACK_CM	R/W	0h	NACK Compatibility mode 0: I2CSTR.NACK bit and NACK interrupt gets triggered when NACK is received in Controller Transmitter mode. 1: I2CSTR.NACK bit and NACK interrupt gets triggered when NACK is received in both Controller Transmitter mode and Target Transmitter mode Reset type: SYSRSn
7-4	SCLKEY	R/W	0h	0xA: Key value which needs to be written to enable Clock (SCL) stretching in ExtendedAuto/Manual mode. Any other value than 0xA: Clock (SCL) stretching mode cannot be enable for both Extended Auto/Manual mode Reset type: SYSRSn
3	MCS	R/W	0h	Manual Clock Stretching mode 0: Manual Clock Stretching is disabled and SCL line is pulled high. 1: Manual Clock Stretching is enabled and SCL line is pulled low allowing software to control clock stretching. Note: When SCLKEY is not 0xA, Manual Clock Stretching mode will remain disabled. Reset type: SYSRSn
2	ECS	R/W	0h	Extended Automatic Clock Stretching mode 0: Extended Automatic Clock stretching feature is disabled 1: Extended Automatic Clock stretching feature is enabled. When enabled, I2C target automatically clock stretches after every ACK / NACK cycle. Note: When SCLKEY is not 0xA, Extended Automatic Clock stretching mode will remain disabled. Reset type: SYSRSn
1	FCM	R/W	0h	Forward Compatibility mode. This bit when programmed brings the functionality of Tx request only when Tx data required regardless of data status in Tx buffer for non-FIFO mode. This register affects the XRDY behavior hence needs to be set after releasing the IRS (I2CMDR[5]). Reset type: SYSRSn

Table 37-22. I2CEMDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	BC	R/W	1h	Backwards compatibility mode. This bit affects the timing of the transmit status bits (XRDY and XSMT) in the I2CSTR register when in TARGET transmitter mode. Reset type: SYSRSn 0h (R/W) = See the 'Backwards Compatibility Mode Bit, TARGET Transmitter' Figure for details. 1h (R/W) = See the 'Backwards Compatibility Mode Bit, TARGET Transmitter' Figure for details.

37.7.2.13 I2CPSC Register (Offset = 18h) [Reset = 0000h]

I2CPSC is shown in Figure 37-33 and described in Table 37-23.

Return to the Summary Table.

The I2C prescaler register (I2CPSC) is a 16-bit register (see Figure 14-21) used for dividing down the I2C input clock to obtain the desired module clock for the operation of the I2C module. See the device-specific data manual for the supported range of values for the module clock frequency.
IPSC must be initialized while the I2C module is in reset (IRS = 0 in I2CMDR). The prescaled frequency takes effect only when IRS is changed to 1. Changing the IPSC value while IRS = 1 has no effect.

Figure 37-33. I2CPSC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
IPSC							
R/W-0h							

Table 37-23. I2CPSC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	IPSC	R/W	0h	I2C prescaler divide-down value. IPSC determines how much the CPU clock is divided to create the module clock of the I2C module: module clock frequency = I2C input clock frequency/(IPSC + 1) Note: IPSC must be initialized while the I2C module is in reset (IRS = 0 in I2CMDR). Reset type: SYSRSn

37.7.2.14 I2CFFTX Register (Offset = 40h) [Reset = 0000h]

I2CFFTX is shown in Figure 37-34 and described in Table 37-24.

Return to the Summary Table.

The I2C transmit FIFO register (I2CFFTX) is a 16-bit register that contains the I2C FIFO mode enable bit as well as the control and status bits for the transmit FIFO mode of operation on the I2C peripheral.

Figure 37-34. I2CFFTX Register

15	14	13	12	11	10	9	8
RESERVED	I2CFFEN	TXFFRST	TXFFST				
R-0h	R/W-0h	R/W-0h	R-0h				
7	6	5	4	3	2	1	0
TXFFINT	TXFFINTCLR	TXFFIENA	TXFFIL				
R-0h	R-0/W1S-0h	R/W-0h	R/W-0h				

Table 37-24. I2CFFTX Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	I2CFFEN	R/W	0h	I2C FIFO mode enable bit. This bit must be enabled for either the transmit or the receive FIFO to operate correctly. Reset type: SYSRSn 0h (R/W) = Disable the I2C FIFO mode. 1h (R/W) = Enable the I2C FIFO mode.
13	TXFFRST	R/W	0h	Transmit FIFO Reset Reset type: SYSRSn 0h (R/W) = Reset the transmit FIFO pointer to 0000 and hold the transmit FIFO in the reset state. 1h (R/W) = Enable the transmit FIFO operation.
12-8	TXFFST	R	0h	Contains the status of the transmit FIFO: xxxxx Transmit FIFO contains xxxxx bytes. 00000 Transmit FIFO is empty. Note: Since these bits are reset to zero, the transmit FIFO interrupt flag will be set when the transmit FIFO operation is enabled and the I2C is taken out of reset. This will generate a transmit FIFO interrupt if enabled. To avoid any detrimental effects from this, write a one to the TXFFINTCLR once the transmit FIFO operation is enabled and the I2C is taken out of reset. Reset type: SYSRSn
7	TXFFINT	R	0h	Transmit FIFO interrupt flag. This bit cleared by a CPU write of a 1 to the TXFFINTCLR bit. If the TXFFIENA bit is set, this bit will generate an interrupt when it is set. Reset type: SYSRSn 0h (R/W) = Transmit FIFO interrupt condition has not occurred. 1h (R/W) = Transmit FIFO interrupt condition has occurred.
6	TXFFINTCLR	R-0/W1S	0h	Transmit FIFO Interrupt Flag Clear Reset type: SYSRSn 0h (R/W) = Writes of zeros have no effect. Reads return a 0. 1h (R/W) = Writing a 1 to this bit clears the TXFFINT flag.
5	TXFFIENA	R/W	0h	Transmit FIFO Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disabled. TXFFINT flag does not generate an interrupt when set. 1h (R/W) = Enabled. TXFFINT flag does generate an interrupt when set.

Table 37-24. I2CFFTX Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	TXFFIL	R/W	0h	Transmit FIFO interrupt level. These bits set the status level that will set the transmit interrupt flag. When the TXFFST4-0 bits reach a value equal to or less than these bits, the TXFFINT flag will be set. This will generate an interrupt if the TXFFIENA bit is set. Because the I2C on this device has a 16-level transmit FIFO, these bits cannot be configured for an interrupt of more than 16 FIFO levels. Reset type: SYSRSn

37.7.2.15 I2CFFRX Register (Offset = 42h) [Reset = 0000h]

I2CFFRX is shown in Figure 37-35 and described in Table 37-25.

Return to the Summary Table.

The I2C receive FIFO register (I2CFFRX) is a 16-bit register that contains the control and status bits for the receive FIFO mode of operation on the I2C peripheral.

Figure 37-35. I2CFFRX Register

15	14	13	12	11	10	9	8
RESERVED		RXFFRST	RXFFST				
R-0h		R/W-0h	R-0h				
7	6	5	4	3	2	1	0
RXFFINT	RXFFINTCLR	RXFFIENA	RXFFIL				
R-0h	R-0/W1S-0h	R/W-0h	R/W-0h				

Table 37-25. I2CFFRX Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	RESERVED	R	0h	Reserved
13	RXFFRST	R/W	0h	I2C receive FIFO reset bit Reset type: SYSRSn 0h (R/W) = Reset the receive FIFO pointer to 0000 and hold the receive FIFO in the reset state. 1h (R/W) = Enable the receive FIFO operation.
12-8	RXFFST	R	0h	Contains the status of the receive FIFO: xxxxx Receive FIFO contains xxxxx bytes 00000 Receive FIFO is empty. Reset type: SYSRSn
7	RXFFINT	R	0h	Receive FIFO interrupt flag. This bit cleared by a CPU write of a 1 to the RXFFINTCLR bit. If the RXFFIENA bit is set, this bit will generate an interrupt when it is set Reset type: SYSRSn 0h (R/W) = Receive FIFO interrupt condition has not occurred. 1h (R/W) = Receive FIFO interrupt condition has occurred.
6	RXFFINTCLR	R-0/W1S	0h	Receive FIFO interrupt flag clear bit. Reset type: SYSRSn 0h (R/W) = Writes of zeros have no effect. Reads return a zero. 1h (R/W) = Writing a 1 to this bit clears the RXFFINT flag.
5	RXFFIENA	R/W	0h	Receive FIFO interrupt enable bit. Reset type: SYSRSn 0h (R/W) = Disabled. RXFFINT flag does not generate an interrupt when set. 1h (R/W) = Enabled. RXFFINT flag does generate an interrupt when set.
4-0	RXFFIL	R/W	0h	Receive FIFO interrupt level. These bits set the status level that will set the receive interrupt flag. When the RXFFST4-0 bits reach a value equal to or greater than these bits, the RXFFINT flag is set. This will generate an interrupt if the RXFFIENA bit is set. Note: Since these bits are reset to zero, the receive FIFO interrupt flag will be set if the receive FIFO operation is enabled and the I2C is taken out of reset. This will generate a receive FIFO interrupt if enabled. To avoid this, modify these bits on the same instruction as or prior to setting the RXFFRST bit. Because the I2C on this device has a 16-level receive FIFO, these bits cannot be configured for an interrupt of more than 16 FIFO levels. Reset type: SYSRSn

Chapter 38

Power Management Bus Module (PMBus)

This chapter describes the features and operation of the Power Management Bus (PMBus) module.

38.1 Introduction

The PMBus module provides an interface between the microcontroller and devices compliant with the SMI Forum PMBus Specification Part I version 1.0 and Part II version 1.1. The PMBus is based on SMBus, which uses a similar physical layer to the I2C. This chapter assumes you are familiar with the PMBus, SMBus, and I2C bus specifications.

38.1.1 PMBUS Related Collateral

Foundational Materials

- Seven things to know about PMBus (Video)

Getting Started Materials

- C28x PMBus Communications Stack User's Guide Application Report
- *Software Implementation of PMBus over I2C for TMS320F2803x Application Report*

Expert Materials

- 9 things you need to know about PMBus Point-of-Load Power (Video)

38.1.2 Features

The PMBus module has the following features:

- Compliance with the SMI Forum PMBus Specification (Part I v1.0 and Part II v1.1)
- Support for controller and target modes
- Ability to support Zero Hold Time (mentioned in SMBus3.0 specs)
- 5V Fail safe IO and 20mA sink current support. Refer to the device data sheet for more details
- Ability to support VIH = 1.35V. Refer to the device data sheet for more details
- Support for I2C modes
- Support for three speeds:
  - Standard Mode: Up to 100kHz
  - Fast Mode: Up to 400kHz
  - Fast Mode Plus: Up to 1MHz
- Packet error checking
- CONTROL and ALERT signals
- Clock high and low time-outs
- Four-byte transmit and receive buffers
- One maskable interrupt, which can be generated by several conditions:
  - Receive data ready
  - Transmit buffer empty
  - Target address received
  - End of message
  - ALERT input asserted
  - Clock low time-out
  - Clock high time-out
  - Bus free

38.1.3 Block Diagram

Figure 38-1 shows the block diagram for PMBus. The PMBus module handles the lower levels of the PMBus protocol. In addition to controlling signal levels and timing, parsing addresses, and buffering data, the PMBus module also directly supports complex transactions such as Read Word and Process Call.

There are four PMBus signals:

- **SCL** is the bus clock. SCL is normally controlled by the controller, but can be held low by a target to delay a transaction and allow more time for processing.
- **SDA** is the bidirectional data line.
- **CONTROL** is a target input that can trigger an interrupt. CONTROL can be used to shut down a target device.
- **ALERT** is a target output/controller input that allows a target to request attention from the controller.

The SDA and SCL timings produced by the module are derived from SYSCLK. To comply with the PMBus timing specs, the bit clock divider must be set by way of the PMBTIMCLK register to provide a bit clock with respect to the module's configured speed.

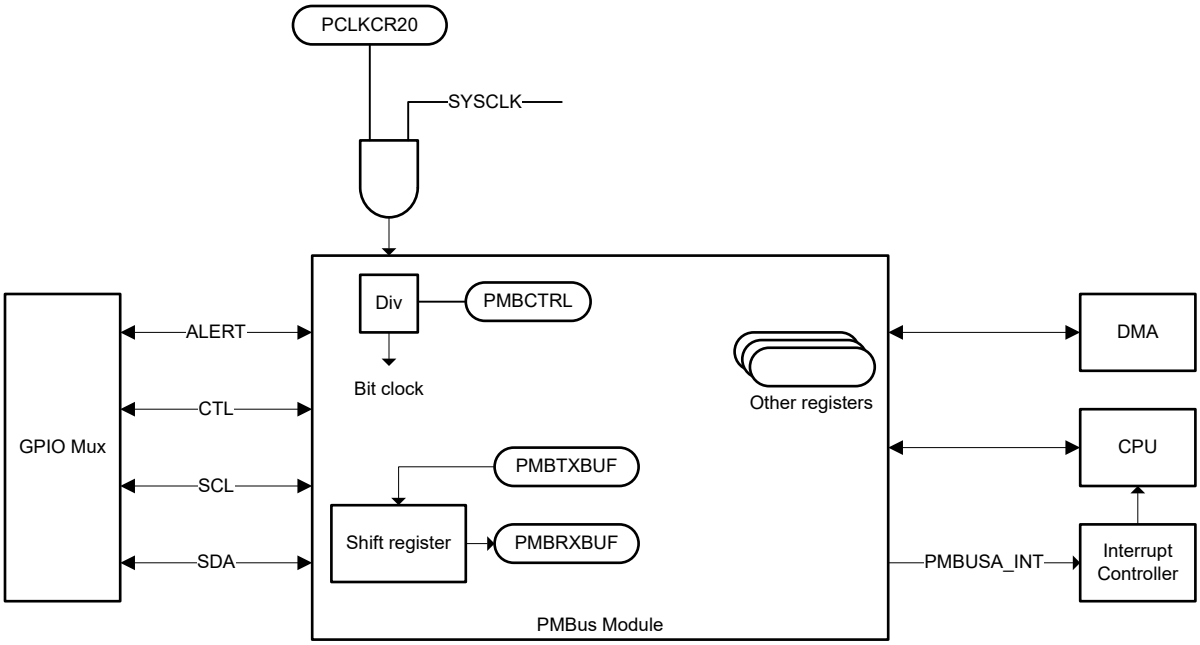


Figure 38-1. PMBus Module Block Diagram

38.2 Configuring Device Pins

The GPIO mux registers must be configured to connect this peripheral to the device pins. To avoid glitches on the pins, the GPyGMUX bits must be configured first (while keeping the corresponding GPyMUX bits at the default of zero), followed by writing the GPyMUX register to the desired value.

Some IO functionality is defined by GPIO register settings independent of this peripheral. For input signals, the GPIO input qualification is set to asynchronous mode by setting the appropriate GPxQSELn register bits to 11b. The internal pullups are configured in the GPyPUD register.

Note

The GPIO configuration register GPyODR must be set to normal mode when the PMBus is used. The open-drain operation for PMBus is managed by the PMBus module

To support a wider range of PMBus IO levels, certain GPIOs have configurable fail-safe systems, V_{IH} minimum thresholds, and configurable sinking capabilities. ***Specifically, the configurable sink current support is mandated for fast plus mode.***

See the *General-Purpose Input/Output (GPIO)* chapter for more details on GPIO mux and settings.

38.3 Target Mode Operation

This section describes the configuration and operation of the PMBus module in target mode.

38.3.1 Configuration

To configure the module, write a clock divider to the PMBCTRL register's CLKDIV field to produce a bit clock frequency with respect to the module's configured speed. To activate target mode, set the TARGET_EN bit in the PMBCTRL register. Next, set up the PMBTCR register. The following options are configurable:

- Target address and mask (TARGET_ADDR and TARGET_MASK): Sets the target address and mask for message acceptance.
- Manual target address acknowledgment (MAN_TARGET_ACK): When enabled, allows software to decide whether to acknowledge (ACK) an address. When disabled, the decision to ACK is made automatically based on the target address and mask.
- PEC enable (PEC_ENA): Set this bit if Packet Error Checking (PEC) is used on the bus.
- Manual command byte acknowledgment (MAN_CMD): Similar to manual target acknowledgment, setting this bit allows software to decide whether to acknowledge (ACK) a command byte.
- Number of bytes to acknowledge automatically (RX_BYTE_ACK_CNT): This is normally set to the maximum value, which allows the entire receive buffer to be used. However, smaller values can be used if the application requires that erroneous messages be detected and not acknowledged (NACKed) as soon as possible.

Manual acknowledgment is done by writing a one to the PMBACK register. Even with automatic acknowledgment, some writes to PMBACK are required. If the message (not including the address) is longer than 4 bytes, each packet of 4 bytes much be acknowledged. The PMBus module stretches the clock (hold the clock low) until an ACK is issued. The module then pulls the data line low and releases the clock, providing the ACK signal to the controller.

If the complete message or the last part of the message is less than 4 bytes (or the RX_BYTE_ACK_CNT limit), do not write to PMBACK.

Writing a zero to the PMBACK bit sends a NACK. This can only be done when the module is waiting for an acknowledgment. If a zero is written at any other time, the NACK is issued during the next message.

38.3.2 Message Handling

This section describes some of the message types for PMBus and how to determine which message type is being received in target mode. This section is oriented toward the most efficient mode of operation – with automatic address and command acknowledgment and is also oriented toward having Packet Error Checking (PEC) enabled.

If automatic address acknowledgment is disabled, all messages start by setting the TARGET_ADDR_READY bit high. Read commands have two instructions one for the read and one for the write. If automatic command acknowledgment is enabled, the DATA READY bit is set high, as well. If the message has no PEC, the number of bytes available is n-1. For example with PEC, a QUICK COMMAND has one byte. With no PEC, a QUICK COMMAND has zero bytes.

Note that the byte count does not increment as bytes arrive. No bits are set in the PMBST register until a stop message is received, the receive buffer is full, or a fault occurs. Then, all appropriate bit values are placed in the register together. All that is necessary to receive a quick command is to ACK the message by writing a one to the PMBACK register.

38.3.2.1 Quick Command

Quick Commands (Figure 38-2) received by the PMBus module in target mode require a simple acknowledgment of the received device address. In automatic address acknowledge mode, the module processes the quick command without firmware interaction. Upon receipt of the end of message, the firmware has the option to read the received address in the PMBHTA register. In manual address acknowledge mode, the address is acknowledged by writing to the PMBACK register.

Figure 38-2. Quick Command Message

38.3.2.2 Send Byte

A Send Byte message (Figure 38-3) consists of the device address, a single data byte, and an optional PEC byte. To process the PEC byte correctly, PEC processing must be enabled in the PMBTCR register. In automatic address acknowledge mode, the data and optional PEC byte are acknowledged without firmware interaction. The module generates an End of Message interrupt, reads the status register and finds the data ready indication bit set. In manual mode, the address is acknowledged by the firmware, while the remaining data and PEC bytes are acknowledged by the module.

The PMBus module stores Data Byte #0 into the PMBRXBUF register. The data byte is stored into bits 7-0. In non-PEC mode, the RX Byte Count in the PMBSTS register indicates one byte received. If PEC processing is enabled, the PEC byte is also stored into the PMBRXBUF register, with the PEC byte residing in bits 15-8. The RX Byte Count in the PMBSTS register indicates two bytes received. The PEC Valid bit in the PMBSTS register indicates the validity of the received PEC byte.

When a Send Byte message is received, the Data Ready bit is set along with the EOM and, assuming that the PEC is valid, the PEC valid bit. The read byte count (RD_BYTE_COUNT) register contains a 2. All that is necessary to receive a send byte command is to ACK the message by writing a 1 to the PMBACK register. Before doing the ACK, read the byte from the lowest byte of the PMBRXBUF register.

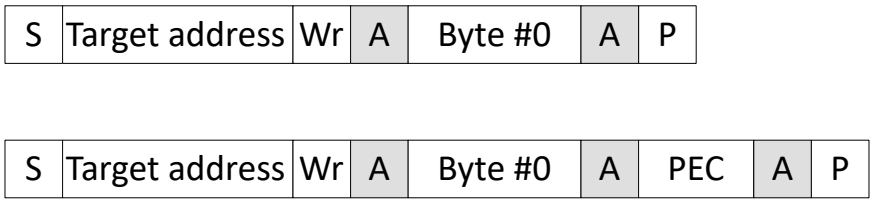


Figure 38-3. Send Byte Message With and Without PEC

38.3.2.3 Receive Byte

A Receive Byte message (Figure 38-4) consists of the device address, a single data byte, and an optional PEC byte. In automatic address acknowledge mode, the firmware receives a data request interrupt following reception of the target address. The data byte to be sent to the controller is stored into bits 7-0 of the PMBTXBUF register and Transmit Byte Count bits within the PMBTCR register are set to a value of 1. If PEC processing is enabled, the Transmit PEC bit (bit 19) within the PMBTCR register is set to 1, along with the Enable PEC bit (bit 15). The module automatically appends the calculated PEC byte at the completion of the message.

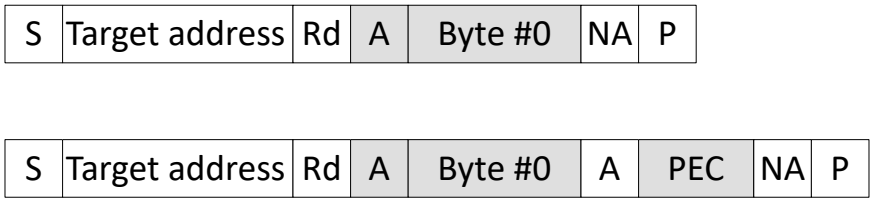


Figure 38-4. Receive Byte Message With and Without PEC

38.3.2.4 Write Byte and Write Word

The Write Byte and Write Word messages (Figure 38-5) consist of a target address, a command word, transmitted data bytes and an optional PEC byte. In automatic address acknowledge mode, the data bytes and optional PEC byte are acknowledged without firmware interaction. The acknowledgment of the command word is configured through the PMBTCR register. The firmware receives an End of Message interrupt in all cases except for Write Word with PEC message, reads the status register and finds the data ready indication bit set.

In the case of a Write Word with PEC byte message, the data ready interrupt is enabled after receiving 4 bytes (command byte, the 2 data bytes and the PEC byte). The firmware reads the data from the PMBRXBUF register and must write the PMBACK register to acknowledge back to the controller. The PMBus module holds SCL low until the firmware responds to the received data.

In all other cases, the EOM interrupt is received and data can be read from the PMBRXBUF register. The firmware is not required to send an acknowledgment back to the controller.

The Write Byte message looks exactly the same as the Send Byte, except the RD_BYTE_COUNT register contains a 3. The Write Word message has a RD_BYTE_COUNT of 4.

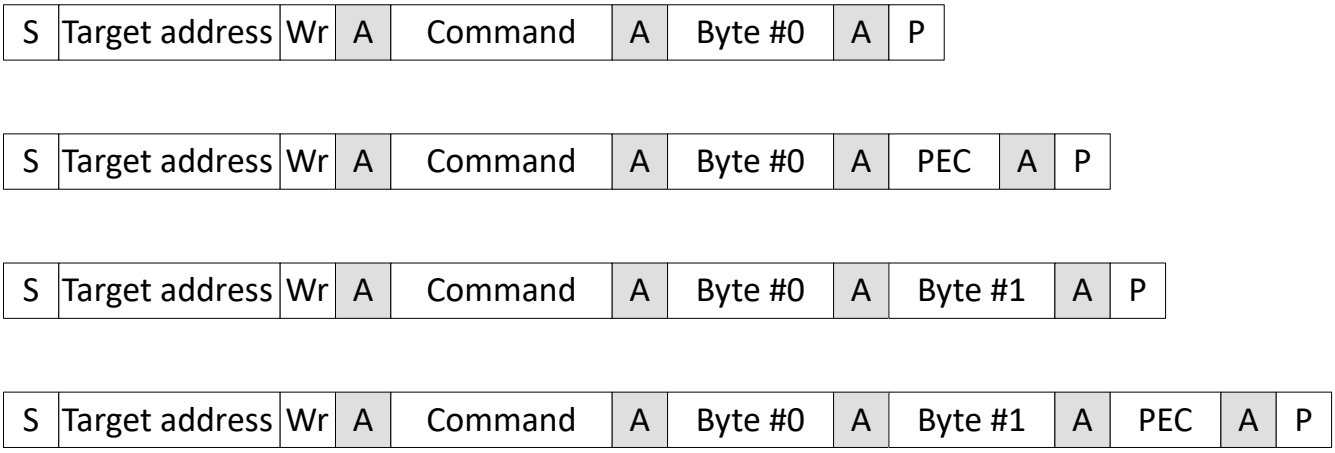


Figure 38-5. Write Byte and Write Word Messages With and Without PEC

38.3.2.5 Read Byte and Read Word

The Read Byte and Read Word messages (Figure 38-6) consist of a target address, a command word, received data bytes from a target, and an optional PEC byte. Address and command acknowledgment is configured through the PMBTCR register. In automatic mode, the PMBus module provides a data ready and data request interrupt following receipt of a repeated start and target address. The received command byte is found in bits 7-0 of the PMBRXBUF register. The firmware responds to the data request by programming the data bytes into the PMBTXBUF register and the TX Byte Count bits in the PMBTCR register. If PEC processing is enabled, the Transmit PEC bit must also be asserted. An EOM interrupt indicates completion of the message to the Controller.

When the repeated start (Sr) signal is received, the Data Ready bit is asserted with a RD_BYTE_COUNT of 1. At this point, the operation cannot be distinguished from a group command Send Byte message. When the same device address is sent out with a read, the Data Request bit is asserted. If data has already been written to the PMTXBUF register before the Device Address is received, the Data Request bit is not asserted. So if group commands are also expected, read the Data Ready with a RD_BYTE_COUNT of 1, and then wait and see whether the next event is an EOM or a Data Request. If the event is an EOM, the command must be processed as a group send byte. If the event is a Data Request, the command must be processed as a read. Depending on the command, the event can be a read byte, word, or block. If the PMBus module is polled, both the Data Ready and the Data Request bits can possibly be set between polling intervals. This must be considered in the design of the firmware.

Once the read command is recognized, you must respond by writing data to the PMBTXBUF register. Make sure that the values in the PMBTCR register are correct. The transmit byte count and PEC bit must be set appropriately. For a read byte, the transmit byte count can be loaded with a 1. If the transmission of a PEC byte is desired, the TX_PEC bit must be set. After this, the data can be written to PMBTXBUF, which starts the transmission. All bytes must be written to PMBTXBUF at the same time. After the controller receives the message, the controller NACKs the last byte to indicate that the correct number of bytes have been received. This causes the EOM bit to be set in the PMBSTS register, indicating to the firmware that the Read Byte message is complete.

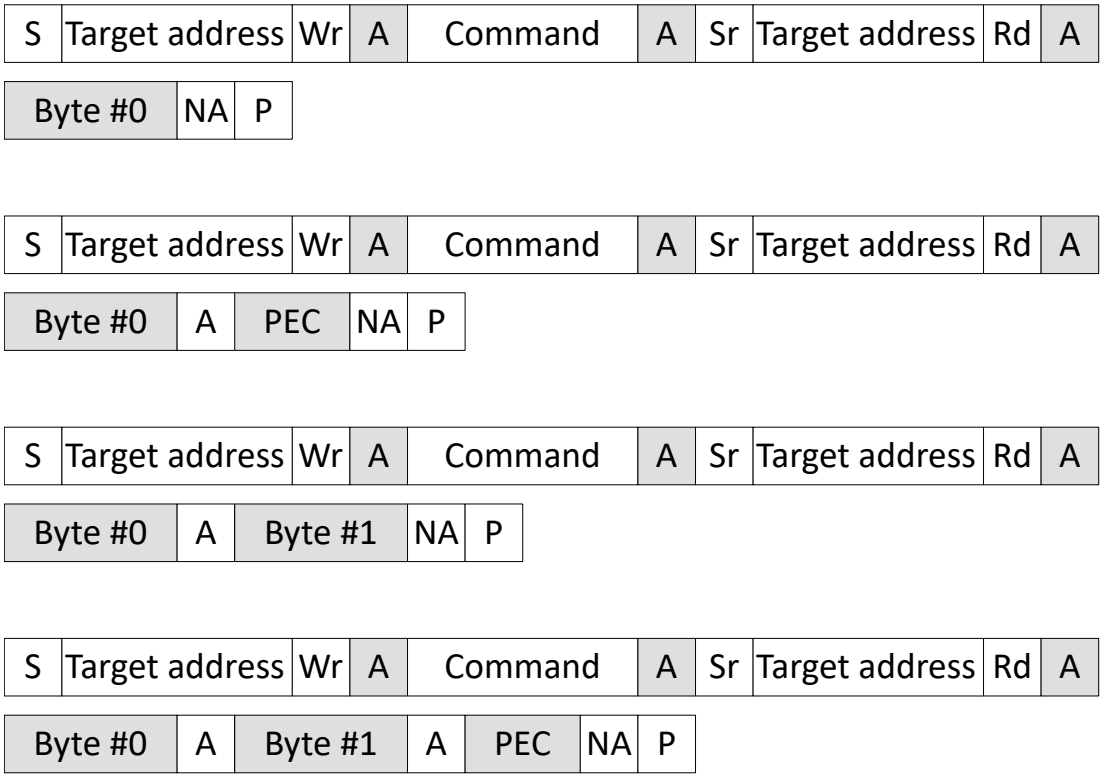


Figure 38-6. Read Byte and Read Word Messages With and Without PEC

38.3.2.6 Process Call

The Process Call (Figure 38-7) protocol consists of a Write Word message, followed by a Read Word message, without a stop condition between the two messages. Address and command acknowledgment is configured through the PMBTCR register. In automatic mode, following receipt of the repeated start and target address, the PMBus module provides a data ready and a data request interrupt. The repeated start bit is set in the PMBSTS register to indicate the receipt of the first part of the Process Call message. The received command byte is found in bits 7-0 of the PMBRXBUF register, while the two data bytes received from the controller can be found in bits 23-8. Upon receipt of the repeated start and a data request from the module, the firmware programs the PMBTXBUF with the 2 data bytes to be sent to the controller. If PEC processing is enabled, the Transmit PEC bit within the PMBTCR register is asserted. The EOM interrupt indicates the read word portion of the Process Call message has been completed by the module.

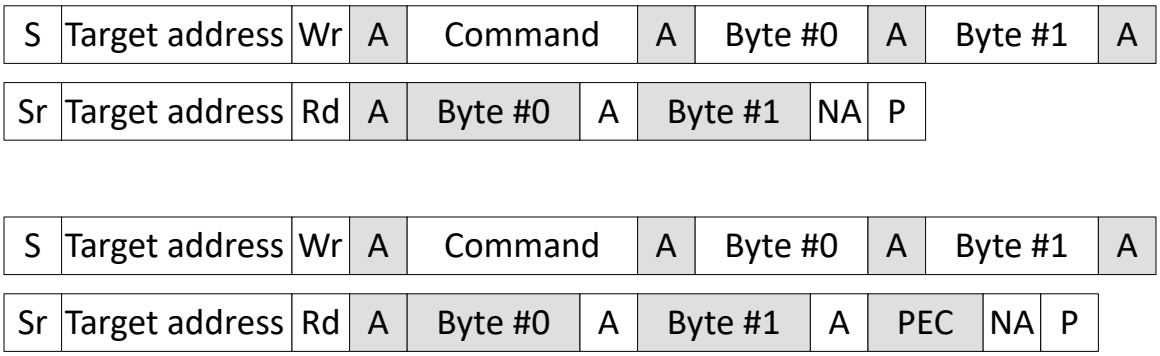


Figure 38-7. Process Call Message With and Without PEC

38.3.2.7 Block Write

The Block Write (Figure 38-8) protocol is similar to Write Word in structure, except that there are more than 2 data bytes in the message. Following the receipt of the command byte, the block length and 2 data bytes, the PMBus module provides a data ready interrupt. The module waits for the firmware to read the received data and program the acknowledge register. While waiting for an ACK from the firmware, the module drives the clock line low, stalling the bus. The data ready interrupts continue for the duration of the message at a frequency of every 4 data bytes. The number of bytes received can be found within the PMBSTS register. At the end of the message, less than 4 bytes can be stored in the PMBRXBUF register. The PEC Valid bit can be checked to determine if the received PEC value is accurate.

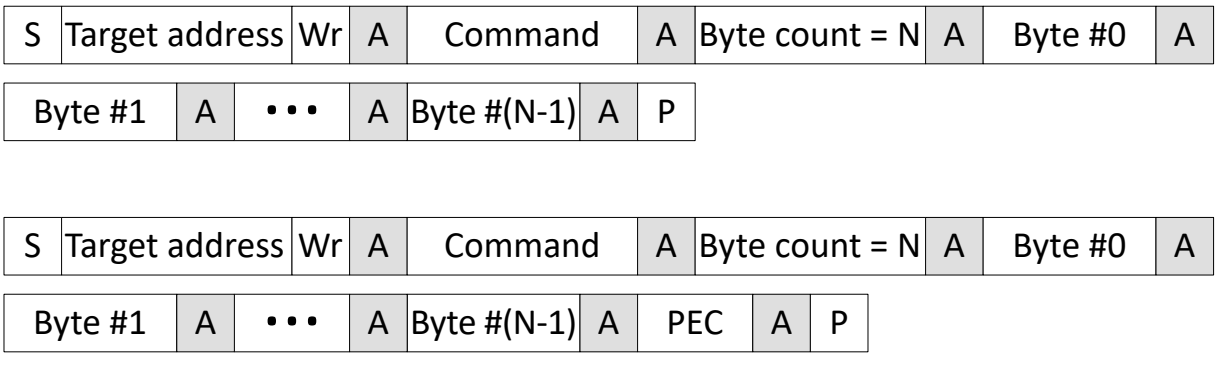


Figure 38-8. Block Write Message With and Without PEC

38.3.2.8 Block Read

The Block Read (Figure 38-9) protocol is similar to a Read Word in structure, except that there are more than 2 data bytes in the message. Following the receipt of the repeated target address, a data ready and data request interrupt is generated by the PMBus module. The command byte received from the controller can be found in bits 7-0 of the PMBRXBUF register. The SCL line is held low until the firmware programs data bytes into the PMBTXBUF register. The firmware is required to load the block length into bits 7-0 of the PMBTXBUF register during the initial programming of the register. After 4 bytes have been transmitted, the module issues a data request interrupt and holds SCL low again until the firmware has programmed additional data into the PMBTXBUF register.

Block read starts the same as Read Word or Read Byte, but TX_COUNT is loaded with a 4 the first time, and TX_PEC is not set. Instead of waiting for an EOM after the first transmission, the firmware instead waits for a Data Request, indicating that the controller is ready for more data. Until the last 4 or less bytes, the firmware simply writes a 4 to TX_COUNT and then writes the 4 bytes to PMBTXBUF. TX_PEC is left cleared. Then when the last 4 or fewer bytes are to be transmitted, the firmware writes out the appropriate byte count, sets the TX_PEC bit, and writes the data to PMBTXBUF. The PMBus module writes out the data, followed by the PEC, and then the EOM bit is set when the controller NACKs the PEC.

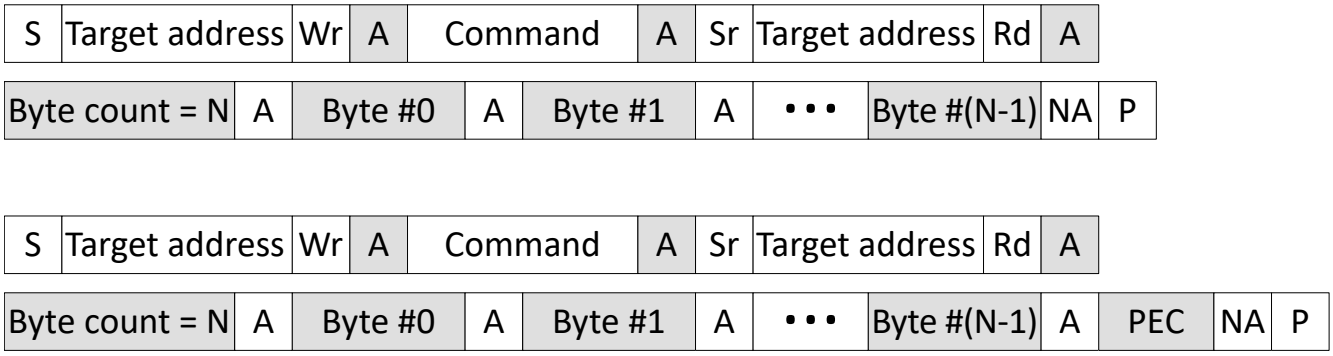


Figure 38-9. Block Read Message With and Without PEC

38.3.2.9 Block Write-Block Read Process Call

The Block Write-Block Read Process Call (Figure 38-10) protocol combines the Block Write and Block Read protocols, removing the stop condition between the two messages. The processing of the Block Read-Block Write Process Call message is similar to the mode of operation for the Process Call message. After acknowledgment of the address and command bytes, the PMBus module generates a data ready interrupt upon detection of 4 data bytes or a repeated start condition. After receiving the repeated start, the firmware is required to load transmit data to send to the controller. Bits 7-0 of the initial programming of the PMBTXBUF register must represent the byte count of the block data sent to the controller.

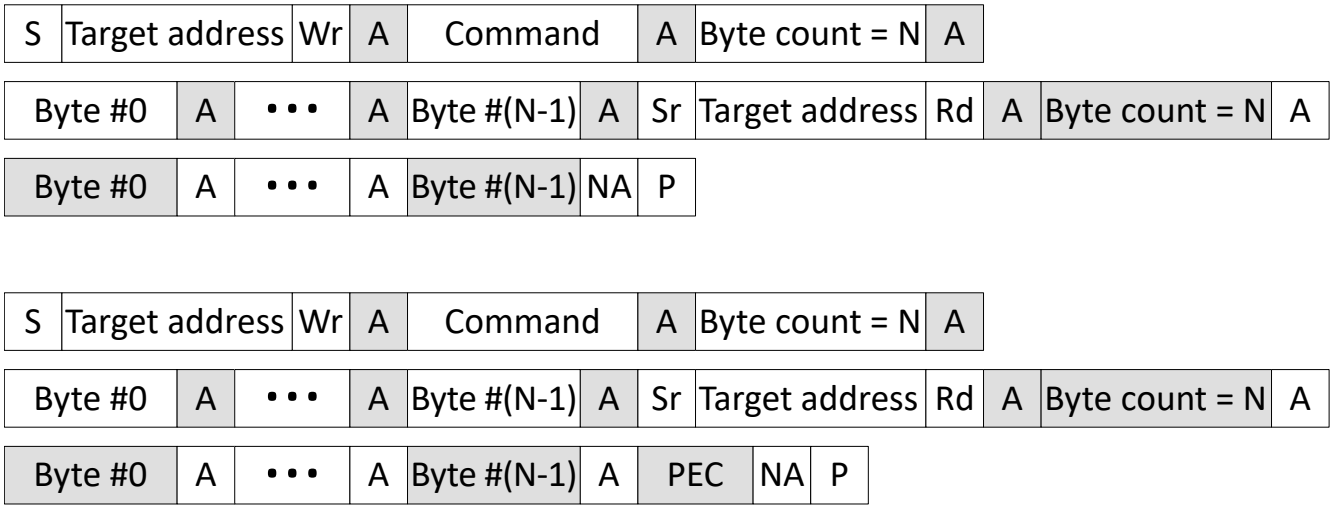


Figure 38-10. Block Write-Block Read Process Call Message With and Without PEC

38.3.2.10 Alert Response

The Alert Response Message (Figure 38-11) is utilized when the controller detects an alert condition from a target on the PMBus. In automatic address acknowledge mode, upon detection of the Alert Response Address, the PMBus module provides an acknowledgment to the controller and sends the programmed target address within the PMBTCR register. The module only responds to the message if the Alert En bit within PMBCTRL register has been previously set. After receiving the Alert Response message, the module clears the alert condition and the enable bit within the PMBCTRL register.

In manual address acknowledge mode, the firmware must read the received address from the PMBRXBUF register and transmit the desired target address back to the controller. The PMBCTRL register must be reprogrammed to disable the Alert En bit used to initiate the Alert Response message from the controller.

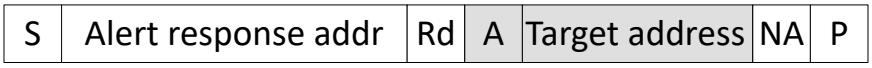


Figure 38-11. Alert Response Message

38.3.2.11 Extended Command

The PMBus module provides support for extended commands that allow for an extra 256 command codes. Both command bytes are stored in the PMBRXBUF register along with the data bytes. In recognizing the extended command messages, the Repeated Start bit and the Rd Byte Count Bits within the PMBSTS register are utilized. For Extended Command Write Byte and Write Word messages (Figure 38-12), the two command bytes are stored in bits 15-0 of the PMBRXBUF register. The initial command byte must hold the command extension code, representing utilization of the extended command protocol. The Repeated Start bit is also set, received after the retransmission of the device address. The Rd Byte Count equals 3 for an Ext Cmd Write Byte message and 4 for an Ext Cmd Write Word message.

For the Extended Command Read Byte and Read Word messages (Figure 38-13), the module generates a data ready and data request interrupt following reception of the repeated device address. The two command bytes are found in bits 15-0 of the PMBRXBUF register, with the initial command byte matching the command extension code. The firmware is required to load transmit data to complete the message back to the controller.

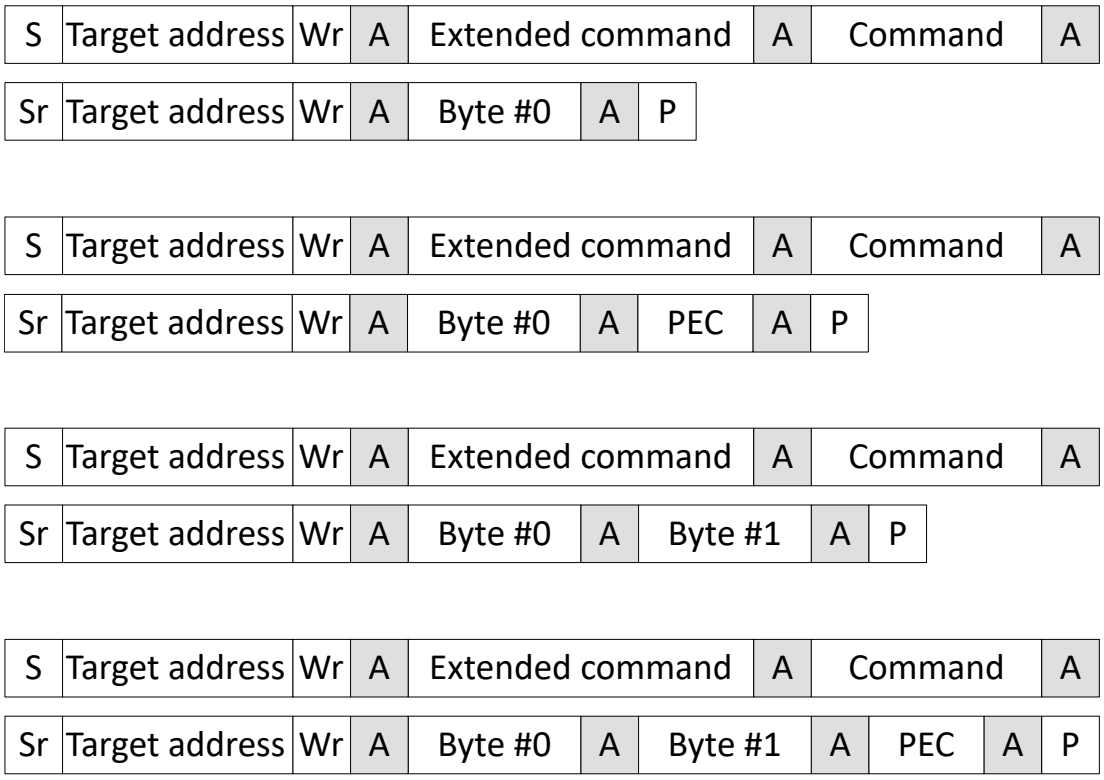


Figure 38-12. Extended Command Write Byte and Write Word Messages With and Without PEC

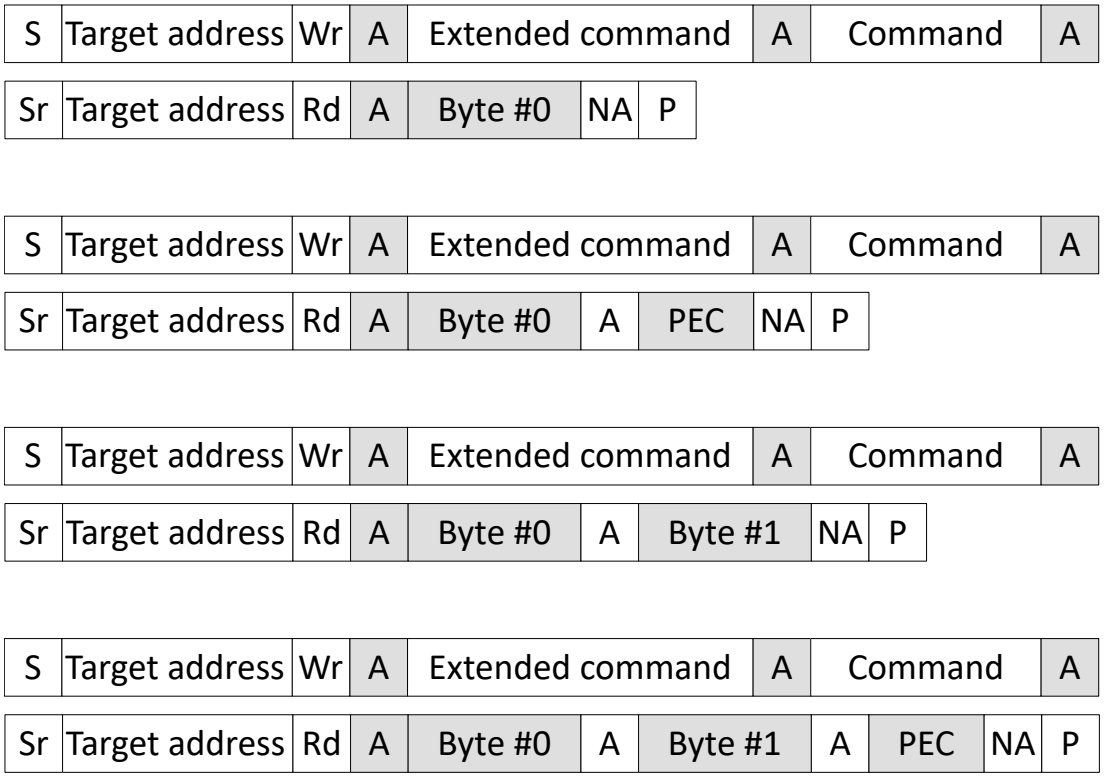


Figure 38-13. Extended Command Read Byte and Read Word Messages With and Without PEC

38.3.2.12 Group Command

The PMBus module supports the Group Command protocol. The Group Command (Figure 38-14) protocol is used to send commands to more than one device within the same message. When devices on the bus detect the stop condition at the conclusion of the Group Command message, the received commands are executed concurrently. Following address and command acknowledgment, the module provides a data ready interrupt upon detection of 4 data bytes or the transmission of a repeated start on the bus. The firmware must wait for the EOM interrupt before processing the received command, as required by the use of the Group Command message.

For Group Commands, the data ready bit is set as soon as the repeated start is received. The data can then be read into memory. But the data must not be acted upon until the EOM bit is set, which occurs when all of the messages have been received. Other than this delayed EOM, there is no difference for the target firmware in receiving a Group Command than any other write message.

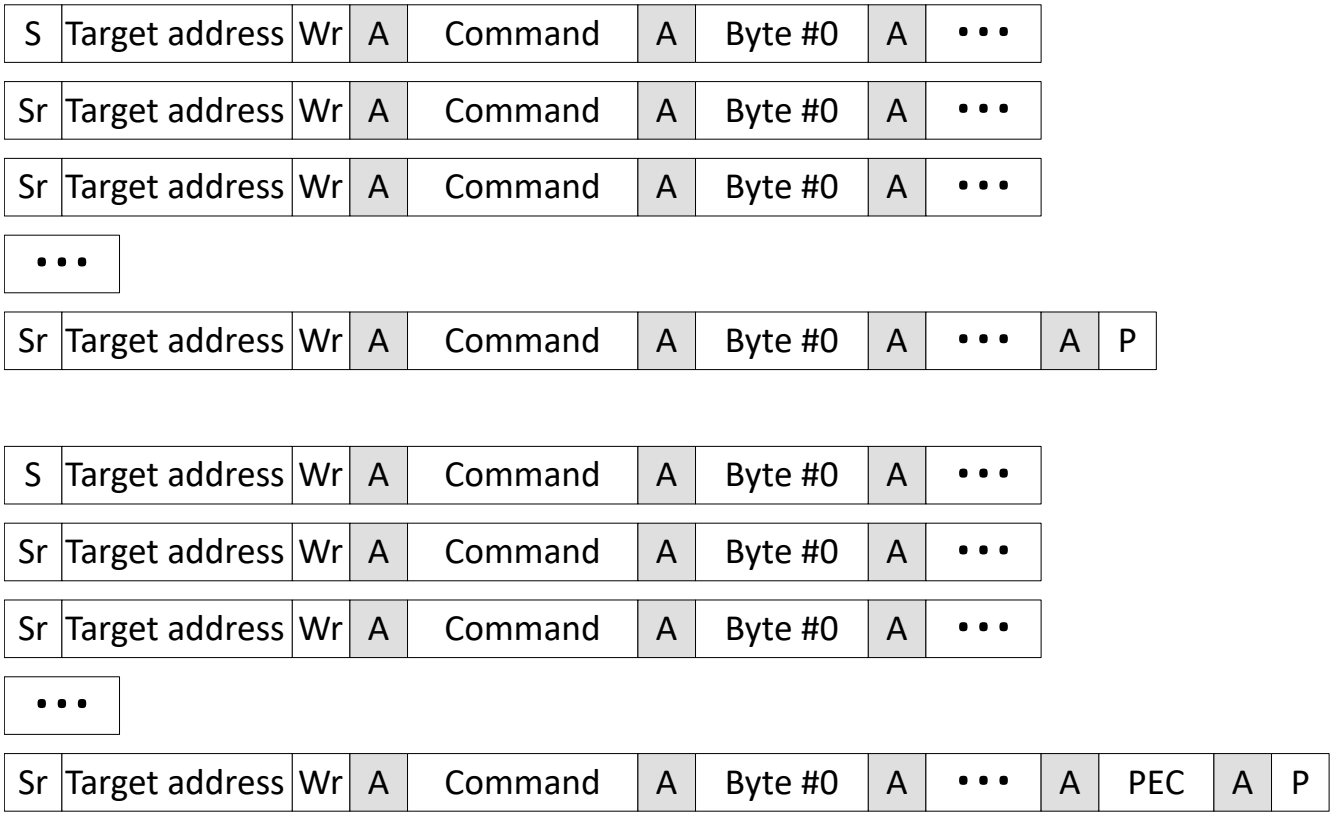


Figure 38-14. Group Command Message With and Without PEC

38.4 Controller Mode Operation

This section describes the configuration and operation of the PMBus module in controller mode.

38.4.1 Configuration

First, write a clock divider to the PMBCTRL register CLKDIV field to produce a bit clock frequency with respect to the module's configured speed. To activate controller mode, set the CONTROLLER_EN bit and clear the TARGET_EN bit in the PMBCTRL register. For each transaction, set up the PMBCCR register. The following options are configurable:

- Target address (TARGET_ADDR): Sets the target address for the next transaction.
- PEC enable (PEC_ENA): If Packet Error Checking (PEC) is used on the bus, set this bit.
- Extended command code enable (EXT_CMD): When set, uses two bytes for commands.
- Command code enable (CMD_ENA): When set, sends a command byte at the start of the transaction.
- Byte count (BYTE_COUNT): Determines the number of data bytes to transfer. This does not include the block length byte, which is generated automatically when needed.
- Special command enables (GRP_CMD and PRC_CALL): Enables special behavior for group commands and process calls.

Writing to the PMBCCR register starts a transfer.

Manual acknowledgment of received data is not needed.

38.4.2 Message Handling

This section describes the behavior and required configuration for each command type.

38.4.2.1 Quick Command

Quick commands (Figure 38-15) are initiated in controller mode by simply programming the desired target device address into the PMBCCR. The byte count within the PMBCCR register is configured to 0 bytes by writing all zeros to bits 15-8. Upon transmission of the device address, the PMBus module monitors the target acknowledgment of the address. If the address is not acknowledged, the NACK bit within the status register is enabled and the PMBus module automatically sends a stop condition on the bus to terminate the message. If the address is acknowledged, a data request is issued to the processor. The firmware writes a zero to the PMBACK to terminate the message, forcing the PMBus modules to write a stop condition onto the bus.

Figure 38-15. Quick Command Message

38.4.2.2 Send Byte

A Send Byte message (Figure 38-16) consists of the device address, a single data byte, and an optional PEC byte. To initiate a Send Byte message, the data byte to be transmitted to the target is loaded into bits 7-0 of the PMBTXBUF register. The PMBCCR register is configured with the device address. To transmit a PEC byte with the message, the PEC_EN bit within the PMBCCR register is asserted high when the address is programmed.

After programming the PMBCCR register, the PMBus module transmits the Send Byte message. The firmware can wait for an End of Message interrupt from the PMBus module. Upon receipt of the EOM interrupt, the PMBSTS register is read to verify the target properly acknowledged the transmitted data.

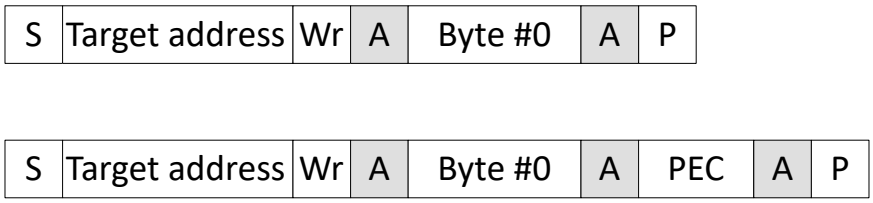


Figure 38-16. Send Byte Message With and Without PEC

38.4.2.3 Receive Byte

A Receive Byte message (Figure 38-17) consists of the device address, a single data byte, and an optional PEC byte. Data is being read from the target in a Receive Byte message. To initiate a Receive Byte message, the firmware programs the device address, the R/W bit and the optional PEC_EN into the PMBCCR register. The R/W bit is enabled high to indicate a read message type (data transmitted from target to controller).

After programming the PMBCCR register, the PMBus module transmits the Receive Byte message. The firmware can wait for an End of Message interrupt from the PMBus module to verify the accuracy of the message transmission. Upon receipt of the EOM interrupt, the PMBSTS register is read to verify proper target acknowledgment of the device address and to determine if any data is available for reading in the PMBRXBUF register. If PEC_EN was asserted in the PMBCCR register, the PEC_VALID bit in the PMBSTS register is also checked to make sure a proper PEC byte was received from the target with the received data.

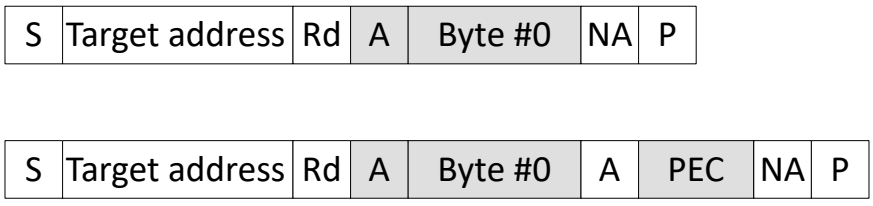


Figure 38-17. Receive Byte Message With and Without PEC

38.4.2.4 Write Byte and Write Word

The Write Byte and Write Word messages (Figure 38-18) consist of a device address, a command byte, transmitted data bytes, and an optional PEC byte. Write Byte messages include a single byte, while the Write Word messages support transmission of 2 bytes to the corresponding target module. Similar to the Send Byte protocol, the PMBCCR register is configured to send 1 or 2 bytes, the CMD_EN bit is set to enable command byte transmission and the optional PEC_EN bit is set.

With the command byte transmission enabled, the format of the PMBTXBUF register differs from the Send Byte protocol. In bits 7-0, the firmware must program the command byte to be sent to the target. The data bytes are programmed into bits 15-8 and bits 23-16.

After programming the PMBCCR register, the PMBus module transmits the Write Byte/Word message. The firmware can wait for an End of Message interrupt from the module to verify the accuracy of the message transmission. The PMBSTS register indicates if the target acknowledged the message properly.

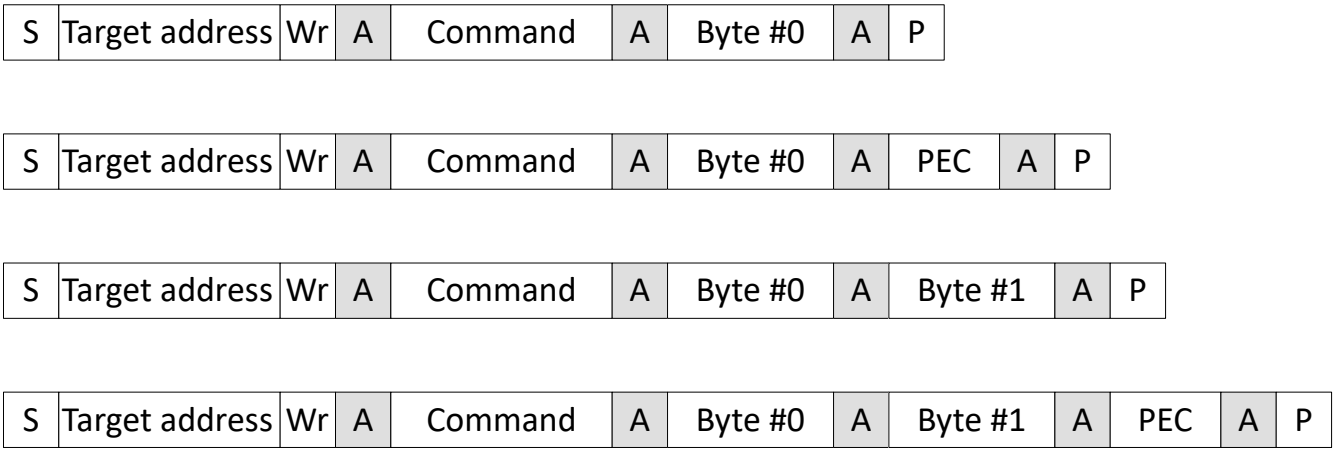


Figure 38-18. Write Byte and Write Word Messages With and Without PEC

38.4.2.5 Read Byte and Read Word

The Read Byte and Read Word messages (Figure 38-19) consist of a device address, a command byte, received data bytes from a target, and an optional PEC byte. Read Byte messages include a single byte, while the Read Word message protocol supports receipt of 2 bytes from the target. Similar to the Receive Byte Protocol, the PMBCCR register is configured to receive 1 or 2 bytes, the CMD_EN bit is set and the PEC_EN is configured to expect or not expect a PEC byte appended to the message. The PMBus module automatically terminates the message after the expected number of bytes is received from the target or if the target does not properly acknowledge any portion of the message.

In addition to programming the PMBCCR register, the firmware is expected to load the command byte into bits 7-0 of the PMBTXBUF register. Any data received from the target is found in the PMBRXBUF register.

Figure 38-19. Read Byte and Read Word Messages With and Without PEC

38.4.2.6 Process Call

The Process Call (Figure 38-20) protocol consists of a Write Word message, followed by a Read Word message, without a stop condition between the two messages. A PEC byte can be appended to the read data from the target as an option to the message protocol. The PMBCCR register includes a PRC_CALL bit, which enables the transmission of a Process Call message onto the PMBus. The PMBus module automatically generates a repeated start condition and initiates the Read Word portion of the message when the process call bit is enabled.

To complete the Write Word portion of the Process Call, the PMBTXBUF register is loaded with the command byte in bits 7-0 and the data bytes are loaded into bits 23-8 of the register.

After programming the PMBCCR register, the PMBus module transmits the Process Call Message. The firmware can wait for an End of Message interrupt from the module to determine the validity of the message. Upon the receipt of the EOM, the PMBSTS register can indicate the receipt of 2 bytes from the Read Word portion of the Process Call message and the status of the target acknowledgment of the transmit data. If PEC processing is enabled, the PEC_VAL bit within the PMBSTS register indicates the accuracy of the PEC byte received from the target during the Read Word part of the message.

The PRC_CALL bit within the PMBCCR register must be disabled for the next non-Process Call message. Note that any write to the PMBCCR register initiates a message, so reconfiguration of the controller is not recommended until the firmware requires a new message to be transmitted.

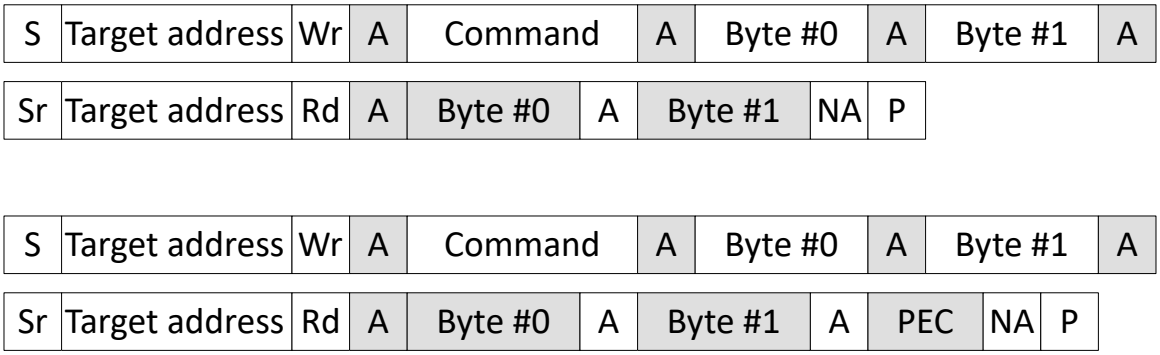


Figure 38-20. Process Call Message With and Without PEC

38.4.2.7 Block Write

The Block Write (Figure 38-21) protocol is similar to a Write Word in structure, with the exception of transmission of more than 2 data bytes in the message. Additionally, the first data byte following the command byte specifies the length of the block of data bytes. As with a majority of the message protocols, the PEC byte can be appended to the end of the write data to the target.

To initiate a Block Write message on the bus, the PMBCCR register is programmed with the block length in the Byte Count bits. The block length is the number of data bytes, excluding the command byte and the first data byte that contains the block length. The PMBus module automatically inserts the block length into the message, if the number of data bytes specified by the firmware exceeds 2. The initial write data is loaded into the PMBTXBUF register. With bits 7-0 representing the command byte, the remaining 3 bytes represent the first 3 data bytes following the block length.

Following programming of the PMBCCR register, the Block Write message is transmitted. If the block length exceeds 3 bytes, the PMBus module provides a data request interrupt, indicating the need for additional data bytes in the PMBTXBUF register. The PMBus module assumes that if more than 4 bytes are needed to complete the message, the firmware utilizes all 4 bytes when programming the PMBTXBUF register. If less than 4 bytes are needed to finish the Block Write message, the firmware only needs to program the appropriate bits of the PMBTXBUF register.

Upon completion of the message, the PMBus module issues an EOM interrupt. The PMBSTS register can be checked to verify the target accepted the block of write data.

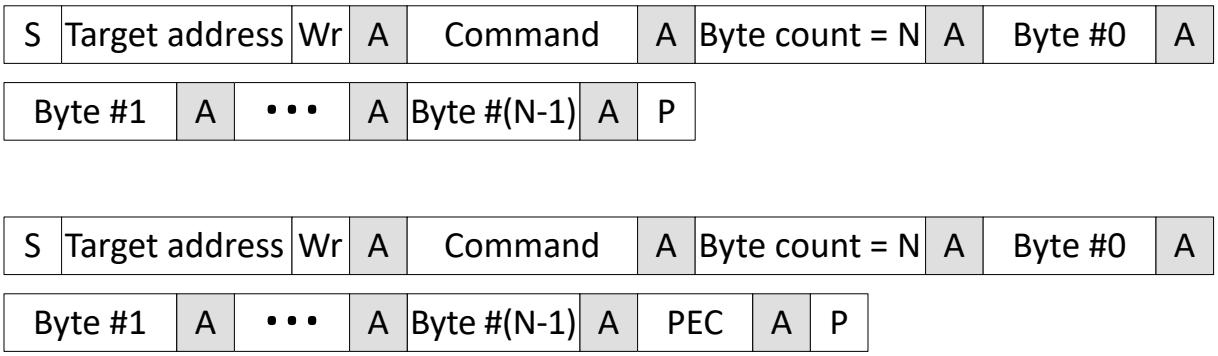


Figure 38-21. Block Write Message With and Without PEC

38.4.2.8 Block Read

The Block Read (Figure 38-22) protocol is similar to a Read Word in structure, with the exception that there are more than 2 data bytes received from the target. The first data byte transmitted by the target represents the block length of the data being written by the target. If PEC processing is enabled, the target appends a PEC byte to the end of the message.

To initiate a Block Read message on the PMBus, the PMBCCR register is programmed with the block length in the Byte Count bits. This count excludes the command byte, any target address and the block length bytes in the message. The command byte to be transmitted to the target is written into bits 7-0 of the PMBTXBUF register prior to the programming of the PMBCCR register.

After configuring the PMBCCR register, the Block Read message is transmitted. The module interrupts the firmware upon receipt of 4 data bytes from the target. If the block length is 3, the EOM interrupt is received concurrently with the data ready interrupt. Otherwise, only a data ready interrupt is asserted, indicating 4 bytes are ready for reading by the firmware. At the end of the message, less than 4 bytes can be stored in the PMBRXBUF register. The RX Byte Count bits in the PMBSTS register indicate the number of bytes available in the final data transfer. The firmware can verify the received PEC upon detection of the End of Message interrupt.

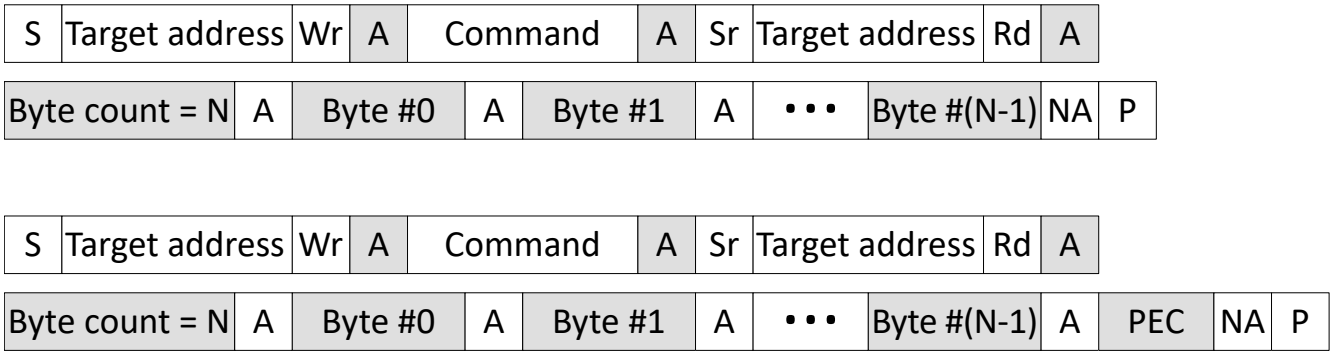


Figure 38-22. Block Read Message With and Without PEC

38.4.2.9 Block Write-Block Read Process Call

The Block Write-Block Read Process Call (Figure 38-23) protocol combines the Block Write and Block Read protocols, removing the stop condition between the two messages. The operation of the controller is similar to a Block Write operation. Loading the block length into the byte count bits of the PMBCCR register provides the length of the Block Write portion of the message. In addition, the PRC_CALL bit within the PMBCCR register must be enabled. Upon completion of the Block Write part of the message, the PMBus module automatically issues a Repeated Start condition on the PMBus and starts transmission of the Block Read portion of the message. Operation of the PMBus module after the Repeated Start condition is the same as a simple Block Read Message.

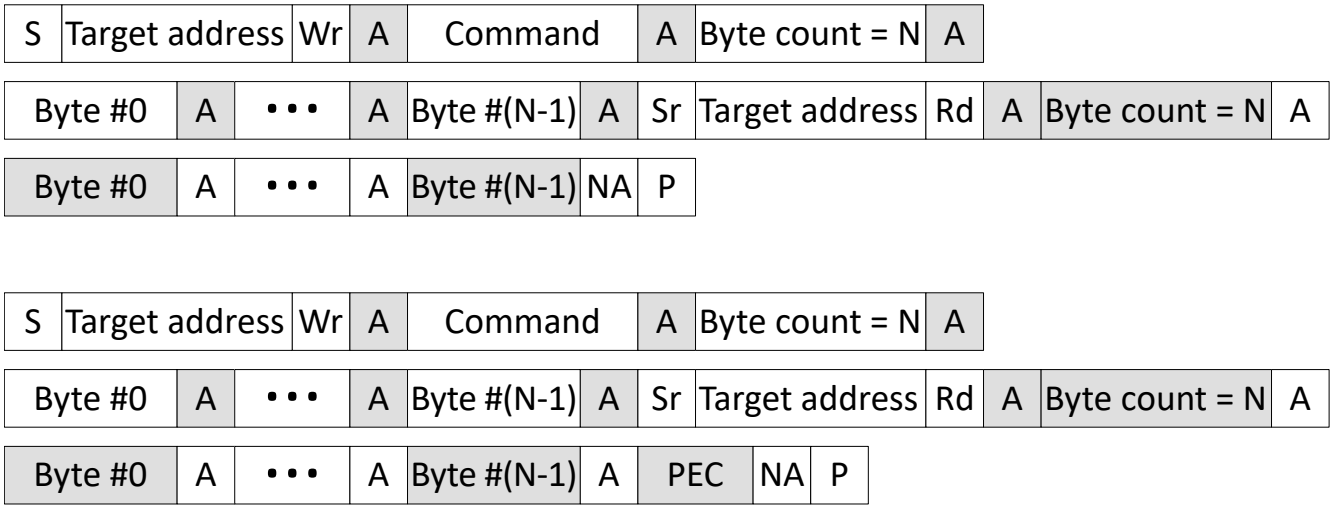


Figure 38-23. Block Write-Block Read Process Call Message With and Without PEC

38.4.2.10 Alert Response

The Alert Response Message (Figure 38-24) is utilized when the controller detects an alert condition from a target. In controller mode, the Alert Response Message is simply a Receive Byte message with PEC disabled and the target address set to 0xC (Alert Response address). The PMBus module detects the alert condition on an input and interrupts the firmware indicating the assertion of an alert condition (target desires to communicate with the controller). Programming the PMBCCR register with the Alert Response address initiates the Alert Response message and provides the device address of the target requesting service. The device address is found in the PMBRXBUF register following receipt of the EOM interrupt.

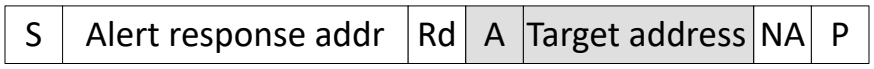


Figure 38-24. Alert Response Message

38.4.2.11 Extended Command

The PMBus module provides support for extended commands which allow for an extra 256 command codes. By asserting the EXT_CMD bit within the PMBCCR register, two command bytes are transmitted on the message protocol. Extended commands can be added to the Write Byte and Write Word (Figure 38-25) and the Read Byte and Read Word (Figure 38-26) protocols. Operation of the PMBus module in extended command mode is similar to these formats. In programming the write data or first part of the read message, the second command byte is loaded into bits 15-8 of the PMBTXBUF register with the remaining data bytes. The remaining operation of the module is identical to the previous protocols, except for the inclusion of a Repeated Start condition and target address in the write messages. No support is required by firmware for these additional bytes in the write messages. The module interprets the EXT_CMD bit and makes the appropriate format changes.

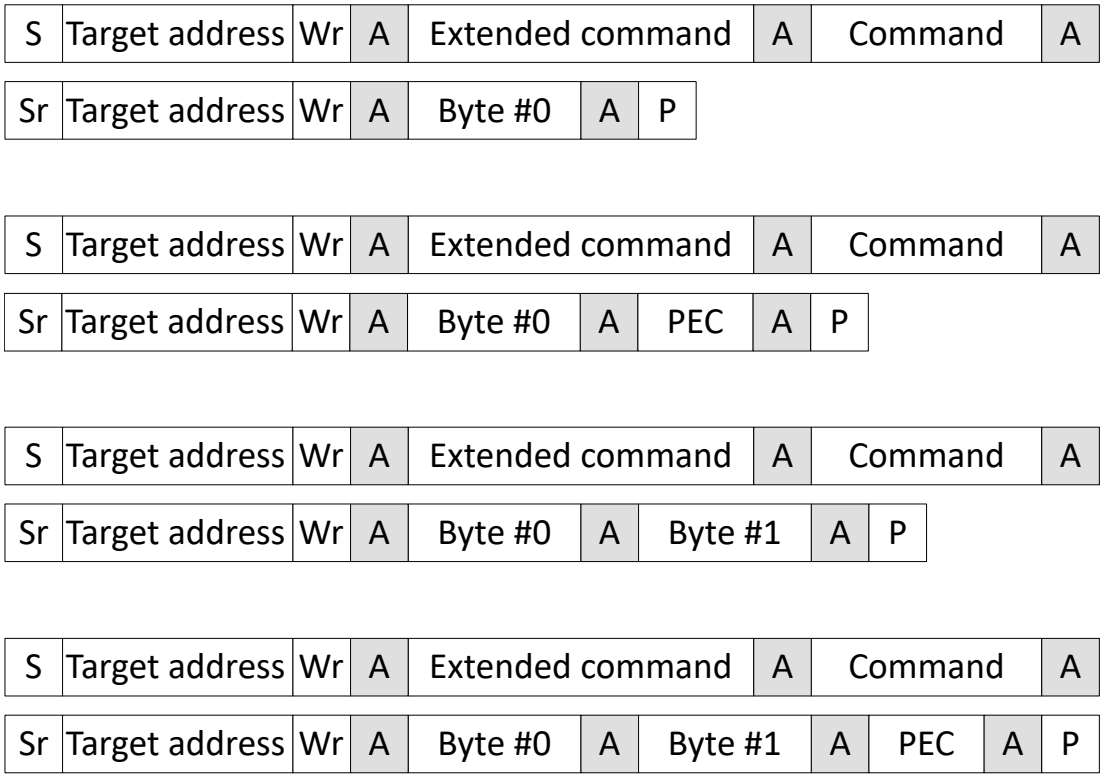


Figure 38-25. Extended Command Write Byte and Write Word Messages With and Without PEC

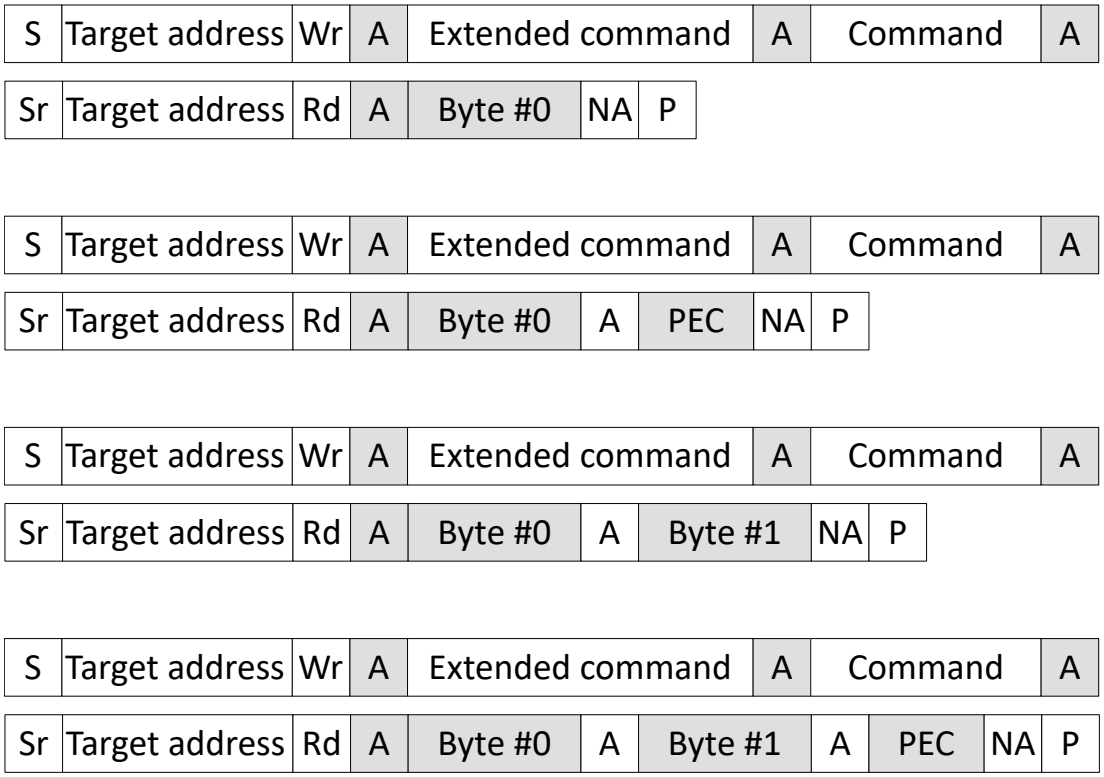


Figure 38-26. Extended Command Read Byte and Read Word Messages With and Without PEC

38.4.2.12 Group Command

The Group Command (Figure 38-27) protocol is used to send commands to more than one device within the same message. When devices on the bus detect the stop condition at the conclusion of the Group Command message, the received commands are executed concurrently. To initiate a Group Command, the GRP_CMD bit within the PMBCCR register must be set when programming the target address for the first device in the message. The rest of the message is processed as a write byte/word message. At the conclusion of the first part of the Group Command message, the firmware programs the next device address in the PMBCCR register. The PMBus module sends a repeated start on the bus and begins the next part of the message. When programming the last device address of the Group Command message, the firmware must disable the GRP_CMD bit when programming the PMBCCR register.

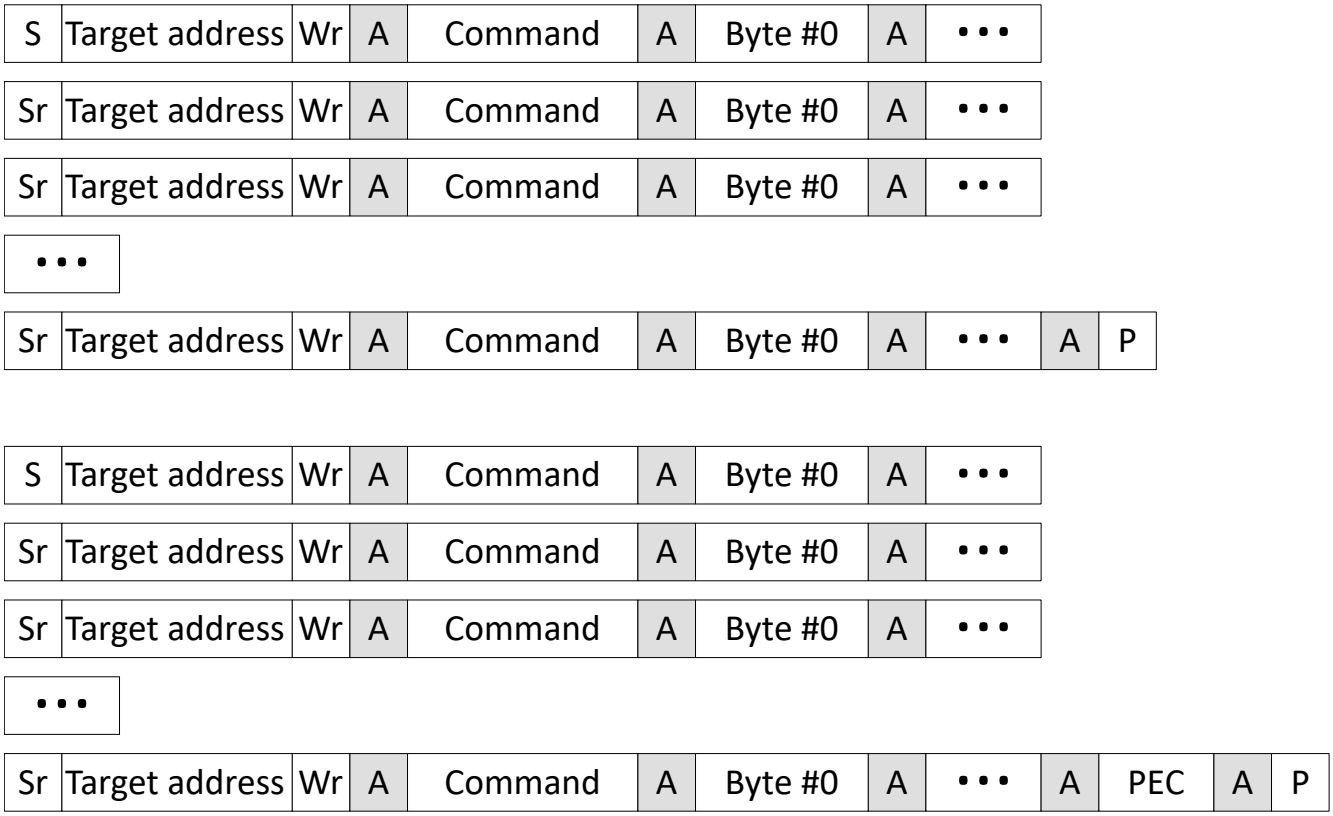


Figure 38-27. Group Command Message With and Without PEC

38.5 Software

38.5.1 PMBUS Registers to Driverlib Functions

Table 38-1. PMBUS Registers to Driverlib Functions

File	Driverlib Function
PMBCCR	
pmbus.h	PMBus_configController
pmbus.h	PMBus_setTargetAddress
PMBTXBUF	
pmbus.c	PMBus_putTargetData
pmbus.c	PMBus_putControllerData
PMBRXBUF	
pmbus.c	PMBus_getData
PMBACK	
pmbus.c	PMBus_ackAddress
pmbus.c	PMBus_ackCommand
pmbus.h	PMBus_ackTransaction
pmbus.h	PMBus_nackTransaction
PMBSTS	
pmbus.c	PMBus_getInterruptStatus
pmbus.h	PMBus_getStatus
PMBINTM	
pmbus.c	PMBus_initTargetMode
pmbus.c	PMBus_configTarget
pmbus.c	PMBus_initControllerMode
pmbus.c	PMBus_configModuleClock
pmbus.c	PMBus_configModuleClockMode
pmbus.c	PMBus_configBusClock
pmbus.h	PMBus_enableInterrupt
pmbus.h	PMBus_disableInterrupt
pmbus.h	PMBus_enableI2CMode
pmbus.h	PMBus_disableI2CMode
PMBTCR	
pmbus.c	PMBus_initTargetMode
pmbus.c	PMBus_configTarget
pmbus.c	PMBus_putTargetData
pmbus.c	PMBus_ackAddress
pmbus.c	PMBus_ackCommand
pmbus.h	PMBus_setOwnAddress
PMBHTA	
pmbus.c	PMBus_verifyPEC
pmbus.h	PMBus_getOwnAddress
pmbus.h	PMBus_getCurrentAccessType
PMBCTRL	
pmbus.c	PMBus_initTargetMode
pmbus.c	PMBus_initControllerMode
pmbus.c	PMBus_configModuleClock

Table 38-1. PMBUS Registers to Driverlib Functions (continued)

File	Driverlib Function
pmbus.c	PMBus_configModuleClockMode
pmbus.c	PMBus_configBusClock
pmbus.h	PMBus_disableModule
pmbus.h	PMBus_enableModule
pmbus.h	PMBus_enableI2CMode
pmbus.h	PMBus_disableI2CMode
pmbus.h	PMBus_assertAlertLine
pmbus.h	PMBus_deassertAlertLine
pmbus.h	PMBus_setCtrlIntEdge
pmbus.h	PMBus_setClkLowTimeoutIntEdge
pmbus.h	PMBus_enableZeroHoldTime
pmbus.h	PMBus_disableZeroHoldTime
PMBTIMCTL	
pmbus.c	PMBus_configBusClock
PMBTIMCLK	
pmbus.c	PMBus_configBusClock
PMBTIMSTSETUP	
pmbus.c	PMBus_configBusClock
PMBTIMBIDLE	
pmbus.c	PMBus_configBusClock
PMBTIMLOWTIMOUT	
pmbus.c	PMBus_configBusClock
PMBTIMHIGHTIMOUT	
pmbus.c	PMBus_configBusClock

38.6 PMBUS Registers

This Section describes the PMBUS Registers.

38.6.1 PMBUS Base Address Table

Table 38-2. PMBUS Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
PMBUS_REGS	PMBUSA_BASE	0x7014_8000	YES	YES	YES	YES	YES	YES	-	YES

38.6.2 PMBUS_REGS Registers

Table 38-3 lists the memory-mapped registers for the PMBUS_REGS registers. All register offset addresses not listed in Table 38-3 should be considered as reserved locations and the register contents should not be modified.

Table 38-3. PMBUS_REGS Registers

Offset	Acronym	Register Name	Protection
0h	PMBCCR	PMBUS CONTROLLER Mode Control Register	
4h	PMBTXBUF	PMBUS Transmit Buffer	
8h	PMBRXBUF	PMBUS Receive buffer	
Ch	PMBACK	PMBUS Acknowledge Register	
10h	PMBSTS	PMBUS Status Register	
14h	PMBINTM	PMBUS Interrupt Mask Register	
18h	PMBTCR	PMBUS TARGET Mode Configuration Register	
1Ch	PMBHTA	PMBUS Hold TARGET Address Register	
20h	PMBCTRL	PMBUS Control Register	
24h	PMBTIMCTL	PMBUS Timing Control Register	
28h	PMBTIMCLK	PMBUS Clock Timing Register	
2Ch	PMBTIMSTSETUP	PMBUS Start Setup Time Register	
30h	PMBTIMBIDLE	PMBUS Bus Idle Time Register	
34h	PMBTIMLOWTIMOUT	PMBUS Clock Low Timeout Value Register	
38h	PMBTIMHIGHTIMOUT	PMBUS Clock High Timeout Value Register	

Complex bit access types are encoded to fit into small table cells. Table 38-4 shows the codes that are used for access types in this section.

Table 38-4. PMBUS_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
RC	R C	Read to Clear
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

38.6.2.1 PMBCCR Register (Offset = 0h) [Reset = 00000000h]

PMBCCR is shown in Figure 38-28 and described in Table 38-5.

Return to the Summary Table.

PMBUS CONTROLLER Mode Control Register

Figure 38-28. PMBCCR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED			PRC_CALL	GRP_CMD	PEC_ENA	EXT_CMD	CMD_ENA
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
BYTE_COUNT							
R/W-0h							
7	6	5	4	3	2	1	0
TARGET_ADDR							RW
R/W-0h							R/W-0h

Table 38-5. PMBCCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R	0h	Reserved
20	PRC_CALL	R/W	0h	0 = Default state for all messages besides Process Call message 1 = Enables transmission of Process Call message Reset type: SYSRSn
19	GRP_CMD	R/W	0h	0 = Default state for all messages besides Group Command message 1 = Enables transmission of Group Command message Reset type: SYSRSn
18	PEC_ENA	R/W	0h	0 = Disables PEC processing 1 = Enables PEC byte transmission/reception Reset type: SYSRSn
17	EXT_CMD	R/W	0h	0 = Use 1 byte for Command Code 1 = Use 2 bytes for Command Code Reset type: SYSRSn
16	CMD_ENA	R/W	0h	0 = Disables use of command code on CONTROLLER initiated messages 1 = Enables use of command code on CONTROLLER initiated messages Reset type: SYSRSn
15-8	BYTE_COUNT	R/W	0h	Indicates number of data bytes transmitted in current message. Byte count does not include any device addresses, command words or block lengths in block messages. In block messages, the PMBus Interface automatically inserts the block length into the message based on the byte count setting. The firmware only needs to load the address, command words and data to be transmitted. PMBus Interface supports byte writes up to 255 bytes. Reset type: SYSRSn

Table 38-5. PMBCCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-1	TARGET_ADDR	R/W	0h	Specifies the address of the TARGET to which the current message is directed towards. Reset type: SYSRSn
0	RW	R/W	0h	0 = Message is a write transaction (data from CONTROLLER to TARGET) 1 = Message is a read transaction (data from TARGET to CONTROLLER) Reset type: SYSRSn

38.6.2.2 PMBTXBUF Register (Offset = 4h) [Reset = 00000000h]

PMBTXBUF is shown in Figure 38-29 and described in Table 38-6.

Return to the Summary Table.

PMBUS Transmit Buffer

Figure 38-29. PMBTXBUF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXDATA																															
R/W-0h																															

Table 38-6. PMBTXBUF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TXDATA	R/W	0h	Bits 31-24: BYTE3 - Last data byte transmitted from Transmit Data Buffer Bits 23-16: BYTE2 - Third data byte transmitted from Transmit Data Buffer Bits 15-8: BYTE1 - Second data byte transmitted from Transmit Data Buffer Bits 7-0: BYTE0 - First data byte transmitted from Transmit Data Buffer Reset type: SYSRSn

38.6.2.3 PMBRXBUF Register (Offset = 8h) [Reset = 00000000h]

PMBRXBUF is shown in Figure 38-30 and described in Table 38-7.

Return to the Summary Table.

PMBUS Receive buffer

Figure 38-30. PMBRXBUF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDATA																															
R-0h																															

Table 38-7. PMBRXBUF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RXDATA	R	0h	Bits 31-24: BYTE3 - Last data byte received in Receive Data Buffer Bits 23-16: BYTE2 - Third data byte received in Receive Data Buffer Bits 15-8: BYTE1 - Second data byte received in Receive Data Buffer Bits 7-0: BYTE0 - First data byte received in Receive Data Buffer Reset type: SYSRSn

38.6.2.4 PMBACK Register (Offset = Ch) [Reset = 00000000h]

PMBACK is shown in Figure 38-31 and described in Table 38-8.

Return to the Summary Table.

PMBUS Acknowledge Register

Figure 38-31. PMBACK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															ACK
R-0h															R/W-0h

Table 38-8. PMBACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	ACK	R/W	0h	0 = NACK received data 1 = Acknowledge received data, bit clears upon issue of ACK on PMBus Reset type: SYSRSn

38.6.2.5 PMBSTS Register (Offset = 10h) [Reset = 00340000h]

PMBSTS is shown in Figure 38-32 and described in Table 38-9.

Return to the Summary Table.

PMBUS Status Register

Figure 38-32. PMBSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED		SCL_RAW	SDA_RAW	CONTROL_RAW	ALERT_RAW	CONTROL_EDGE	ALERT_EDGE
R-0h		R-1h	R-1h	R-0h	R-1h	RC-0h	RC-0h
15	14	13	12	11	10	9	8
CONTROLLER	LOST_ARB	BUS_FREE	UNIT_BUSY	RPT_START	TARGET_ADDR_READY	CLK_HIGH_DETECTED	CLK_LOW_TIMEOUT
RC-0h	RC-0h	RC-0h	RC-0h	RC-0h	RC-0h	RC-0h	RC-0h
7	6	5	4	3	2	1	0
PEC_VALID	NACK	EOM	DATA_REQUEST	DATA_READY	RD_BYTE_COUNT		
RC-0h	RC-0h	RC-0h	RC-0h	RC-0h	RC-0h		

Table 38-9. PMBSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	Reserved
21	SCL_RAW	R	1h	0 = PMBus clock pin observed at logic level low 1 = PMBus clock pin observed at logic level high Reset type: SYSRSn
20	SDA_RAW	R	1h	0 = PMBus data pin observed at logic level low 1 = PMBus data pin observed at logic level high Reset type: SYSRSn
19	CONTROL_RAW	R	0h	0 = Control pin observed at logic level low 1 = Control pin observed at logic level high Reset type: SYSRSn
18	ALERT_RAW	R	1h	0 = Alert pin observed at logic level low 1 = Alert pin observed at logic level high Reset type: SYSRSn
17	CONTROL_EDGE	RC	0h	0 = Control pin has not transitioned 1 = Control pin has been asserted by another device on PMBus Reset type: SYSRSn
16	ALERT_EDGE	RC	0h	0 = Alert pin has not transitioned 1 = Alert pin has been asserted by another device on PMBus Reset type: SYSRSn
15	CONTROLLER	RC	0h	0 = PMBus Interface in TARGET Mode or Idle Mode 1 = PMBus Interface in CONTROLLER Mode Reset type: SYSRSn
14	LOST_ARB	RC	0h	0 = CONTROLLER has attained control of PMBus 1 = CONTROLLER has lost arbitration and control of PMBus Reset type: SYSRSn

Table 38-9. PMBSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	BUS_FREE	RC	0h	0 = PMBus processing current message 1 = PMBus available for new message Reset type: SYSRSn
12	UNIT_BUSY	RC	0h	0 = PMBus Interface is idle, ready to transmit/receive message 1 = PMBus Interface is busy, processing current message Reset type: SYSRSn
11	RPT_START	RC	0h	0 = No Repeated Start received by interface 1 = Repeated Start condition received by interface Reset type: SYSRSn
10	TARGET_ADDR_READY	RC	0h	0 = Indicates no TARGET address is available for reading 1 = TARGET address ready to be read from Receive Data Register (Bits 6:0) Reset type: SYSRSn
9	CLK_HIGH_DETECTED	RC	0h	0 = No Clock High condition detected 1 = Clock High exceeded 50us during message Reset type: SYSRSn
8	CLK_LOW_TIMEOUT	RC	0h	0 = No clock low timeout detected 1 = Clock low timeout detected, clock held low for greater than 35ms Reset type: SYSRSn
7	PEC_VALID	RC	0h	0 = Received PEC not valid (if EOM is asserted) 1 = Received PEC is valid Note: PEC_VALID status is don't care during the message. This will have a valid value only after EOM. Reset type: SYSRSn
6	NACK	RC	0h	0 = Data transmitted has been accepted by receiver 1 = Receiver has not accepted transmitted data Reset type: SYSRSn
5	EOM	RC	0h	0 = Message still in progress or PMBus in idle state. 1 = End of current message detected Reset type: SYSRSn
4	DATA_REQUEST	RC	0h	0 = No data needed by PMBus Interface 1 = PMBus Interface request additional data. PMBus clock stretching enabled to stall bus Reset type: SYSRSn
3	DATA_READY	RC	0h	0 = No data available for reading by processor 1 = PMBus Interface read buffer full, firmware required to read data prior to further bus activity. PMBus clock stretching enabled to stall bus until data is read by firmware. Reset type: SYSRSn
2-0	RD_BYTE_COUNT	RC	0h	0 = No received data 1 = 1 byte received. Data located in Receive Data Register, Bits 7-0 2 = 2 bytes received. Data located in Receive Data Register, Bits 15-0 3 = 3 bytes received. Data located in Receive Data Register, Bits 23-0 4 = 4 bytes received. Data located in Receive Data Register, Bits 31-0 Reset type: SYSRSn

38.6.2.6 PMBINTM Register (Offset = 14h) [Reset = 000003FFh]

PMBINTM is shown in Figure 38-33 and described in Table 38-10.

Return to the Summary Table.

PMBUS Interrupt Mask Register

Figure 38-33. PMBINTM Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						CLK_HIGH_DETECT	LOST_ARB
R-0h						R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
CONTROL	ALERT	EOM	TARGET_ADDR_READY	DATA_REQUEST	DATA_READY	BUS_LOW_TIMEOUT	BUS_FREE
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 38-10. PMBINTM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	Reserved
9	CLK_HIGH_DETECT	R/W	1h	0 = Generates interrupt if clock high exceeds 50us during message 1 = Disables interrupt generation for Clock High detection Reset type: SYSRSn
8	LOST_ARB	R/W	1h	0 = Generates interrupt upon assertion of Lost Arbitration flag 1 = Disables interrupt generation upon assertion of Lost Arbitration flag Reset type: SYSRSn
7	CONTROL	R/W	1h	0 = Generates interrupt upon assertion of Control flag 1 = Disables interrupt generation upon assertion of Control flag Reset type: SYSRSn
6	ALERT	R/W	1h	0 = Generates interrupt upon assertion of Alert flag 1 = Disables interrupt generation upon assertion of Alert flag Reset type: SYSRSn
5	EOM	R/W	1h	0 = Generates interrupt upon assertion of End of Message flag 1 = Disables interrupt generation upon assertion of End of Message flag Reset type: SYSRSn
4	TARGET_ADDR_READY	R/W	1h	0 = Generates interrupt upon assertion of TARGET Address Ready flag 1 = Disables interrupt generation upon assertion of TARGET Address Ready flag Reset type: SYSRSn
3	DATA_REQUEST	R/W	1h	0 = Generates interrupt upon assertion of Data Request flag 1 = Disables interrupt generation upon assertion of Data Request flag Reset type: SYSRSn

Table 38-10. PMBINTM Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DATA_READY	R/W	1h	0 = Generates interrupt upon assertion of Data Ready flag 1 = Disables interrupt generation upon assertion of Data Ready flag Reset type: SYSRSn
1	BUS_LOW_TIMEOUT	R/W	1h	0 = Generates interrupt upon assertion of Clock Low Timeout flag 1 = Disables interrupt generation upon assertion of Clock Low Timeout flag Reset type: SYSRSn
0	BUS_FREE	R/W	1h	0 = Generates interrupt upon assertion of Bus Free flag 1 = Disables interrupt generation upon assertion of Bus Free flag Reset type: SYSRSn

38.6.2.7 PMBTCR Register (Offset = 18h) [Reset = 00607F7Ch]

PMBTCR is shown in Figure 38-34 and described in Table 38-11.

Return to the Summary Table.

PMBUS TARGET Mode Configuration Register

Figure 38-34. PMBTCR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED	RX_BYTE_ACK_CNT		MAN_CMD	TX_PEC	TX_COUNT		
R-0h	R/W-3h		R/W-0h	R/W-0h	R/W-0h		
15	14	13	12	11	10	9	8
PEC_ENA	TARGET_MASK						
R/W-0h	R/W-7Fh						
7	6	5	4	3	2	1	0
MAN_TARGET_ACK	TARGET_ADDR						
R/W-0h	R/W-7Ch						

Table 38-11. PMBTCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	0h	Reserved
22-21	RX_BYTE_ACK_CNT	R/W	3h	Configures number of data bytes to automatically acknowledge when receiving data in TARGET mode. 00 = 1 byte received by TARGET. Firmware is required to manually acknowledge every received byte. 01 = 2 bytes received by TARGET. Hardware automatically acknowledges the first received byte. Firmware is required to manually acknowledge after the second received byte. 10 = 3 bytes received by TARGET. Hardware automatically acknowledges the first 2 received bytes. Firmware is required to manually acknowledge after the third received byte. 11 = 4 bytes received by TARGET. Hardware automatically acknowledges the first 3 received bytes. Firmware is required to manually acknowledge after the fourth received byte Reset type: SYSRSn
20	MAN_CMD	R/W	0h	0 = TARGET automatically acknowledges received command code 1 = Data Request flag generated after receipt of command code, firmware required to issue ACK to continue message Reset type: SYSRSn
19	TX_PEC	R/W	0h	Asserted when the TARGET needs to send a PEC byte at end of message. PMBus Interface will transmit the calculated PEC byte after transmitting the number of data bytes indicated by TX Byte Cnt(Bits 18:16). Reset type: SYSRSn

Table 38-11. PMBTCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18-16	TX_COUNT	R/W	0h	0 = No bytes valid 1 = One byte valid, Byte #0 (Bits 7:0 of Transmit Data Register) 2 = Two bytes valid, Bytes #0 and #1 (Bits 15:0 of Transmit Data Register) 3 = Three bytes valid, Bytes #0-2 (Bits 23:0 of Transmit Data Register) 4 = Four bytes valid, Bytes #0-3 (Bits 31:0 of Transmit Data Register) Reset type: SYSRSn
15	PEC_ENA	R/W	0h	0 = PEC processing disabled 1 = PEC processing enabled Reset type: SYSRSn
14-8	TARGET_MASK	R/W	7Fh	Used in address detection, the TARGET mask enables acknowledgement of multiple device addresses by the TARGET. Writing a '0' to a bit within the TARGET mask enables the corresponding bit in the TARGET address to be either '1' or '0' and still allow for a match. Writing a '0' to all bits in the mask enables the PMBus Interface to acknowledge any device address. Upon power-up, the TARGET mask defaults to 7Fh, indicating the TARGET will only acknowledge the address programmed into the TARGET Address (Bits 6-0). Reset type: SYSRSn
7	MAN_TARGET_ACK	R/W	0h	0 = TARGET automatically acknowledges device address specified in TARGET_ADDR, Bits 6:0 1 = Enables the Manual TARGET Address Acknowledgement Mode. Firmware is required to read received address and acknowledge on every message Note: When bit 31 (I2C_mode) of PMBCTRL register is set it is recommended to use manual acknowledging of TARGET address only (MAN_TARGET_ACK =1). Reset type: SYSRSn
6-0	TARGET_ADDR	R/W	7Ch	Configures the current device address of the TARGET. Used in automatic TARGET address acknowledge mode (default mode). The PMBus Interface will compare the received device address with the value stored in the TARGET Address bits and the mask configured in the TARGET Mask bits. If matching, the TARGET will acknowledge the device address. Reset type: SYSRSn

38.6.2.8 PMBHTA Register (Offset = 1Ch) [Reset = 00000000h]

PMBHTA is shown in Figure 38-35 and described in Table 38-12.

Return to the Summary Table.

PMBUS Hold TARGET Address Register

Figure 38-35. PMBHTA Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TARGET_ADDR							TARGET_RW
R-0h							R-0h

Table 38-12. PMBHTA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-1	TARGET_ADDR	R	0h	Stored device address acknowledged by the TARGET Reset type: SYSRSn
0	TARGET_RW	R	0h	Stored R/W bit from address acknowledged by the TARGET 0 = Write Access 1 = Read Access Reset type: SYSRSn

38.6.2.9 PMBCTRL Register (Offset = 20h) [Reset = 00200000h]

PMBCTRL is shown in Figure 38-36 and described in Table 38-13.

Return to the Summary Table.

PMBUS Control Register

Figure 38-36. PMBCTRL Register

31	30	29	28	27	26	25	24
I2CMODE	ZH_EN	RESERVED		CLKDIV			
R/W-0h	R/W-0h	R-0h		R/W-0h			
23	22	21	20	19	18	17	16
CLKDIV	CONTROLLER_EN	TARGET_EN	CLK_LO_DIS	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	SDA_DIR	SDA_VALUE	SDA_MODE	CNTL_DIR	CNTL_VALUE	CNTL_MODE	ALERT_DIR
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
ALERT_VALUE	ALERT_MODE	CNTL_INT_EDGE	FAST_MODE_PLUS	FAST_MODE	BUS_LO_INT_EDGE	ALERT_EN	RESET
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 38-13. PMBCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I2CMODE	R/W	0h	0 = PMBUS mode 1 = I2C mode Reset type: SYSRSn
30	ZH_EN	R/W	0h	PMBus Zero data Hold Time Enable 0: PMBus Zero Hold Time Disabled. Doesn't support zero hold time mentioned in SMBus3.0 specification. 1: PMBus Zero Hold Time Enable. Supports zero hold time mentioned in SMBus3.0 specification Reset type: SYSRSn
29-28	RESERVED	R	0h	Reserved
27-23	CLKDIV	R/W	0h	The clock to the PMBUS transmit/receive FSMs (FSM_CLK) is divided version of the SYSCLK clock. Frequency(FSM_CLK) = Frequency(SYSCLK)/(CLKDIV+1) Note: FSM_CLK should be less than (or) equal to 10MHz in standard and fast mode. FSM_CLK should be between 20-25MHz in fast mode plus. Reset type: SYSRSn
22	CONTROLLER_EN	R/W	0h	0 = Disables PMBus CONTROLLER capability 1 = Enables PMBus CONTROLLER capability Reset type: SYSRSn
21	TARGET_EN	R/W	1h	0 = Disables PMBus TARGET capability 1 = Enables PMBus TARGET capability Reset type: SYSRSn
20	CLK_LO_DIS	R/W	0h	0 = Clock Low Timeout Enabled 1 = Clock Low Timeout Disabled Reset type: SYSRSn
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved

Table 38-13. PMBCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	SDA_DIR	R/W	0h	0 = PMBus data pin configured as output 1 = PMBus data pin configured as input Reset type: SYSRSn
13	SDA_VALUE	R/W	0h	0 = PMBus data pin driven low in GPIO Mode 1 = PMBus data pin driven high in GPIO Mode Reset type: SYSRSn
12	SDA_MODE	R/W	0h	0 = PMBus data pin driven low in GPIO Mode 1 = PMBus data pin driven high in GPIO Mode Reset type: SYSRSn
11	CNTL_DIR	R/W	0h	0 = Control pin configured as output 1 = Control pin configured as input Reset type: SYSRSn
10	CNTL_VALUE	R/W	0h	0 = Control pin driven low in GPIO Mode 1 = Control pin driven high in GPIO Mode Reset type: SYSRSn
9	CNTL_MODE	R/W	0h	0 = Control pin configured in functional mode (Default) 1 = Control pin configured as GPIO Reset type: SYSRSn
8	ALERT_DIR	R/W	0h	0 = Alert pin configured as output 1 = Alert pin configured as input Reset type: SYSRSn
7	ALERT_VALUE	R/W	0h	0 = Alert pin driven low in GPIO Mode 1 = Alert pin driven high in GPIO Mode Reset type: SYSRSn
6	ALERT_MODE	R/W	0h	0 = Alert pin configured in functional mode 1 = Aler3 pin configured as GPIO Reset type: SYSRSn
5	CNTL_INT_EDGE	R/W	0h	0 = Interrupt generated on falling edge of Control 1 = Interrupt generated on rising edge of Control Reset type: SYSRSn
4	FAST_MODE_PLUS	R/W	0h	0 = Standard 100 KHz mode enabled 1 = Fast Mode Plus enabled (1MHz operation on PMBus) Reset type: SYSRSn
3	FAST_MODE	R/W	0h	0 = Standard 100 KHz mode enabled 1 = Fast Mode enabled (400KHz operation on PMBus) Reset type: SYSRSn
2	BUS_LO_INT_EDGE	R/W	0h	0 = Interrupt generated on rising edge of clock low timeout 1 = Interrupt generated on falling edge of clock low timeout Reset type: SYSRSn
1	ALERT_EN	R/W	0h	0 = PMBus Alert is not driven by TARGET, pulled up high on PMBus 1 = PMBus Alert driven low by TARGET Reset type: SYSRSn
0	RESET	R/W	0h	0 = No reset of internal state machines (Default) 1 = Control state machines are reset to initial states Note: Status register PMBSTS should be explicitly cleared by reading the register after softrest as this will not be cleared by Software Reset. Reset type: SYSRSn

38.6.2.10 PMBTIMCTL Register (Offset = 24h) [Reset = 00000000h]

PMBTIMCTL is shown in Figure 38-37 and described in Table 38-14.

Return to the Summary Table.

PMBUS Timing Control Register

Figure 38-37. PMBTIMCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TIM_OVERRIDE
R-0h							R/W-0h

Table 38-14. PMBTIMCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	TIM_OVERRIDE	R/W	0h	0 PMBUS FSMs uses the default settings of the timing parameters. 1 PMBUS FSMs would use the settings in following registers: * PMBTIMCLK * PMBTIMSTSETUP * PMBTIMBIDLE * PMBTIMLOWTIMOUT * PMBTIMHIGHTIMOUT Reset type: SYSRSn

38.6.2.11 PMBTIMCLK Register (Offset = 28h) [Reset = 0060002Fh]

PMBTIMCLK is shown in Figure 38-38 and described in Table 38-15.

Return to the Summary Table.

PMBUS Clock Timing Register

Figure 38-38. PMBTIMCLK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								CLK_FREQ							
R-0h								R/W-60h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLK_HIGH_LIMIT							
R-0h								R/W-2Fh							

Table 38-15. PMBTIMCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	CLK_FREQ	R/W	60h	Defines the number of PMBUS FSM input clock in the PMBUS CONTROLLER clock period. Number of FSM clocks in the one clock period = (CLK_FREQ+4) Reset type: SYSRSn
15-8	RESERVED	R	0h	Reserved
7-0	CLK_HIGH_LIMIT	R/W	2Fh	Defines the number of PMBUS FSM input clock in the PMBUS CONTROLLER clock high pulse. Number of FSM clocks in the one clock high pulse = (CLK_HIGH_LIMIT+3) Reset type: SYSRSn

38.6.2.12 PMBTIMSTSETUP Register (Offset = 2Ch) [Reset = 0000002Fh]

PMBTIMSTSETUP is shown in Figure 38-39 and described in Table 38-16.

Return to the Summary Table.

PMBUS Start Setup Time Register

Figure 38-39. PMBTIMSTSETUP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								TSU_STA							
R-0h																								R/W-2Fh							

Table 38-16. PMBTIMSTSETUP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	TSU_STA	R/W	2Fh	Determines the Setup time between last rise edge of the PMBUS CONTROLLER clock and the next start edge, TSU_STA value defines the setup time in terms of PMBUS FSM clock cycles. Reset type: SYSRSn

38.6.2.13 PMBTIMBIDLE Register (Offset = 30h) [Reset = 000001F3h]

PMBTIMBIDLE is shown in Figure 38-40 and described in Table 38-17.

Return to the Summary Table.

PMBUS Bus Idle Time Register

Figure 38-40. PMBTIMBIDLE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						BUSIDLE									
R-0h																						R/W-1F3h									

Table 38-17. PMBTIMBIDLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	Reserved
9-0	BUSIDLE	R/W	1F3h	Determines the duration for which PMBUS clock and Data are 1 , to conclude that the bus is IDLE. BUSIDLE value is in terms of number of PMBUS FSM clock cycles. Reset type: SYSRSn

38.6.2.14 PMBTIMLOWTIMOUT Register (Offset = 34h) [Reset = 0005572Fh]

PMBTIMLOWTIMOUT is shown in Figure 38-41 and described in Table 38-18.

Return to the Summary Table.

PMBUS Clock Low Timeout Value Register

Figure 38-41. PMBTIMLOWTIMOUT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												CLKLOWTIMOUT																			
R-0h												R/W-0005572Fh																			

Table 38-18. PMBTIMLOWTIMOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19-0	CLKLOWTIMOUT	R/W	0005572Fh	Determines the duration for which PMBUS clock if low , will result in a clock low timeout condition. CLKLOWTIMOUT value is in terms of number of PMBUS FSM clock cycles. Reset type: SYSRSn

38.6.2.15 PMBTIMHIGHTIMOUT Register (Offset = 38h) [Reset = 000001F3h]

PMBTIMHIGHTIMOUT is shown in Figure 38-42 and described in Table 38-19.

Return to the Summary Table.

PMBUS Clock High Timeout Value Register

Figure 38-42. PMBTIMHIGHTIMOUT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						CLKHIGHTIMOUT									
R-0h						R/W-1F3h									

Table 38-19. PMBTIMHIGHTIMOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	Reserved
9-0	CLKHIGHTIMOUT	R/W	1F3h	Determines the duration for which PMBUS clock if high , will result in a clock high timeout condition. CLKHIGHTIMOUT value is in terms of number of PMBUS FSM clock cycles. Reset type: SYSRSn

Chapter 39

Universal Asynchronous Receiver/Transmitter (UART)

This chapter describes the Universal Asynchronous Receivers/Transmitters (UARTs).

39.1 Introduction

The UART module performs the functions of parallel-to-serial and serial-to-parallel conversions.

39.1.1 Features

The Universal Asynchronous Receiver/Transmitter (UART) module in this device contains the following features:

- Programmable baud-rate generator allowing speeds up to 12.5Mbps for regular speed (divide by 16) and 25Mbps for high speed (divide by 8)
- Separate 16-deep and 8-bit wide transmit (TX) and receive (RX) FIFOs to reduce CPU interrupt service loading
- Programmable FIFO length providing conventional double-buffered interface
- FIFO trigger levels of ⅛, ¼, ½, ¾, and ⅞
- Standard asynchronous communication bits for start, stop, and parity
- Line-break generation and detection
- Fully programmable serial interface characteristics
  - 5, 6, 7, or 8 data bits
  - Even, odd, stick, or no parity bit generation and detection
  - 1 or 2 stop bit generation
- IrDA serial-IR (SIR) encoder and decoder providing
  - Programmable use of IrDA SIR or UART input/output
  - Support of IrDA SIR encoder and decoder functions for data rates up to 115.2kbps half-duplex
  - Support of normal 3/16 and low-power (1.41 to 2.23µs) bit durations
  - Programmable internal clock generator enabling division of reference clock by 1 to 256 for low-power mode bit duration
- EIA-485 9-bit support
- Standard FIFO-level and End-of-Transmission (EOT) interrupts
- Efficient transfers using Real-Time Direct Memory Access Controller (RTDMA)
  - Separate channels for transmit and receive
  - Receive single request asserted when data is in the FIFO; burst request asserted at programmed FIFO level
  - Transmit single request asserted when there is space in the FIFO; burst request asserted at programmed FIFO level
- SYSCLK is used to generate the baud clock.

39.1.2 UART Related Collateral

Foundational Materials

- C29x Academy - Universal Asynchronous Receiver Transmitter (UART)

39.1.3 Block Diagram

Figure 39-1 shows the UART block diagram.

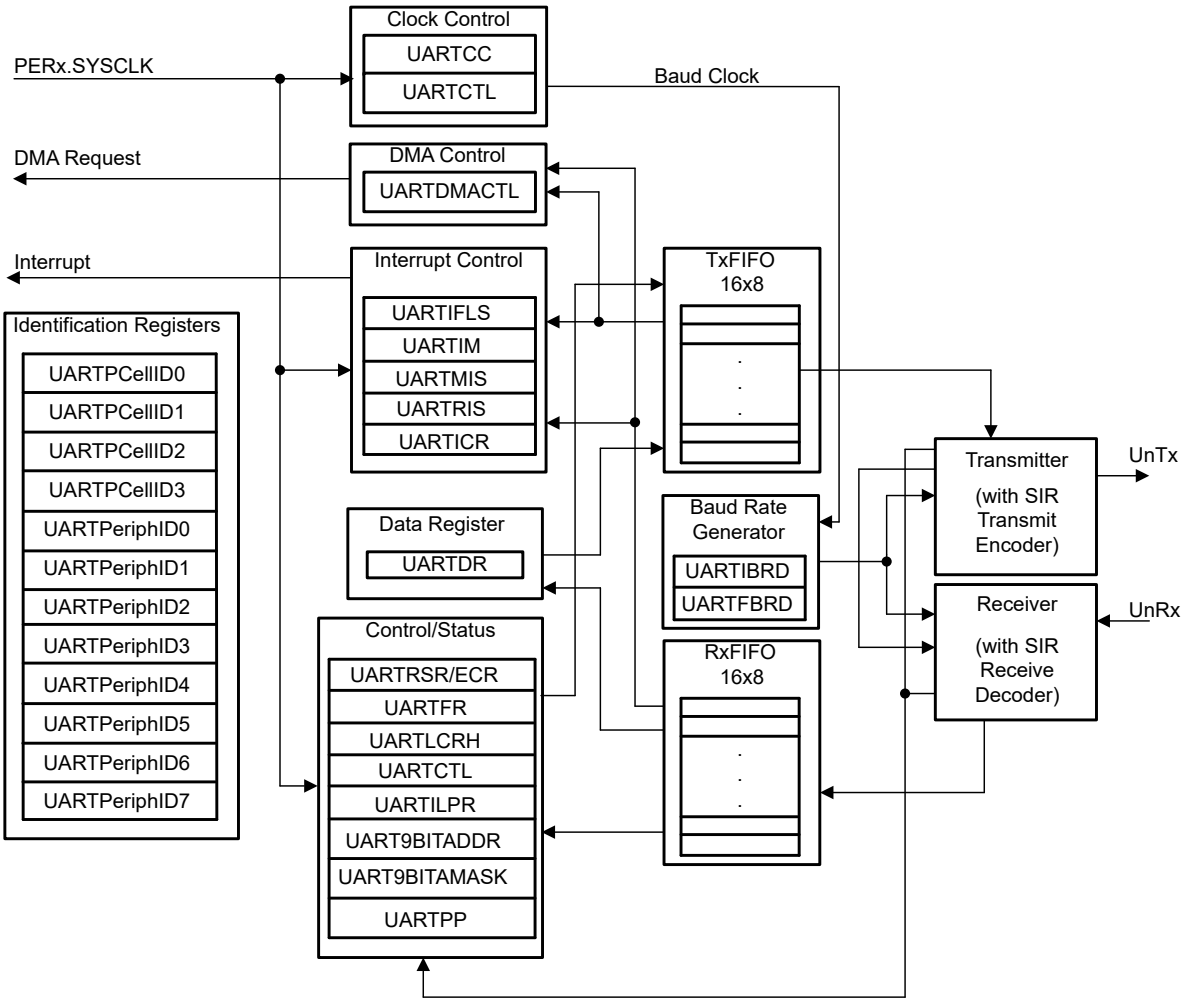


Figure 39-1. UART Module Block Diagram

39.2 Functional Description

The UART module performs the functions of parallel-to-serial and serial-to-parallel conversions. The UART module is similar in functionality to a 16C550 UART, but is not register-compatible.

The UART is configured for transmit or receive through the TXE and RXE bits of the UART Control (UARTCTL) register. Transmit and receive are both enabled out of reset. Before any control registers are programmed, the UART must be disabled by clearing the UARTEN bit in UARTCTL. If the UART is disabled during a TX or RX operation, the current transaction is completed prior to the UART stopping.

The UART module also includes a serial IR (SIR) encoder and decoder block that can be connected to an infrared transceiver to implement an IrDA SIR physical layer. The SIR function is programmed using the UARTCTL register.

39.2.1 Transmit and Receive Logic

The transmit logic performs parallel-to-serial conversion on the data read from the transmit FIFO. The control logic outputs the serial bit stream beginning with a start bit and followed by the data bits (LSB first), parity bit, and the stop bits according to the programmed configuration in the control registers. See Figure 39-2 for details.

The receive logic performs serial-to-parallel conversion on the received bit stream after a valid start pulse has been detected. Overrun, parity, frame error checking, and line-break detection are also performed, and the status accompanies the data that is written to the receive FIFO.

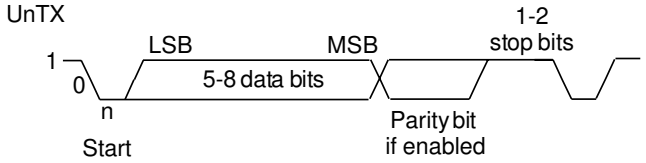


Figure 39-2. UART Character Frame

39.2.2 Baud-Rate Generation

The baud-rate divisor is a 22-bit number consisting of a 16-bit integer and a 6-bit fractional part. The number formed by these two values is used by the baud-rate generator to determine the bit period. Having a fractional baud-rate divisor allows the UART to generate all the standard baud rates.

The 16-bit integer is loaded through the UART Integer Baud-Rate Divisor (UARTIBRD) register and the 6-bit fractional part is loaded with the UART Fractional Baud-Rate Divisor (UARTFBRD) register. The baud-rate divisor (BRD) has the following relationship to the system clock (where BRDI is the integer part of the BRD, and BRDF is the fractional part, separated by a decimal place.)

$$BRD = BRDI + BRDF = UARTSysClk / (ClkDiv \times Baud\ Rate)$$

where

UARTSysClk is the system clock (SYSCLK) connected to the UART, and ClkDiv is either 16 (if HSE in UARTCTL is clear) or 8 (if HSE in UARTCTL is set).

By default, this is the main system clock (SYSCLK).

The 6-bit fractional number (that is to be loaded into the DIVFRAC bit field in the UARTFBRD register) can be calculated by taking the fractional part of the baud-rate divisor, multiplying by 64, and adding 0.5 to account for rounding errors:

$$UARTFBRD[DIVFRAC] = integer(BRDF \times 64 + 0.5)$$

The UART generates an internal baud-rate reference clock at 8× or 16× the baud-rate (referred to as Baud8 and Baud16, depending on the setting of the HSE bit [bit 5] in UARTCTL). This reference clock is divided by 8 or 16 to generate the transmit clock, and is used for error detection during receive operations.

Along with the UART Line Control High Byte (UARTLCRH) register, the UARTIBRD and UARTFBRD registers form an internal 30-bit register. This internal register is only updated when a write operation to UARTLCRH is performed, so any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register for the changes to take effect.

To update the baud-rate registers, there are four possible sequences:

- UARTIBRD write, UARTFBRD write, and UARTLCRH write
- UARTFBRD write, UARTIBRD write, and UARTLCRH write
- UARTIBRD write and UARTLCRH write
- UARTFBRD write and UARTLCRH write

39.2.3 Data Transmission

Data received or transmitted is stored in two 16-byte FIFOs, though the receive FIFO has an extra four bits per character for status information. For transmission, data is written into the transmit FIFO. If the UART is enabled, the UART causes a data frame to start transmitting with the parameters indicated in the UARTLCRH register. Data continues to be transmitted until there is no data left in the transmit FIFO. The BUSY bit in the UART Flag (UARTFR) register is asserted as soon as data is written to the transmit FIFO (that is, if the FIFO is non-empty) and remains asserted while data is being transmitted. The BUSY bit is negated only when the transmit FIFO is empty, and the last character has been transmitted from the shift register, including the stop bits. The UART can indicate that the UART is busy even though the UART is no longer enabled.

When the receiver is idle (the UnRx signal is continuously 1), and the data input goes Low (a start bit has been received), the receive counter begins running and data is sampled on the eighth cycle of Baud16 or fourth cycle of Baud8 depending on the setting of the HSE bit (bit 5) in UARTCTL (described in Section 39.2.1).

The start bit is valid and recognized if the UnRx signal is still low on the eighth cycle of Baud16 (HSE clear) or the fourth cycle of Baud8 (HSE set), otherwise the start bit is ignored. After a valid start bit is detected, successive data bits are sampled on every 16th cycle of Baud16 or eighth cycle of Baud8 (that is, one bit period later) according to the programmed length of the data characters and value of the HSE bit in UARTCTL. The parity bit is then checked if parity mode is enabled. Data length and parity are defined in the UARTLCRH register.

Lastly, a valid stop bit is confirmed if the UnRx signal is High, otherwise a framing error has occurred. When a full word is received, the data is stored in the receive FIFO along with any error bits associated with that word.

39.2.4 Serial IR (SIR)

The UART peripheral includes an IrDA SIR encoder and decoder block. The IrDA SIR block provides functionality that converts between an asynchronous UART data stream and a half-duplex serial SIR interface. No analog processing is performed on-chip. The role of the SIR block is to provide a digital encoded output and decoded input to the UART. When enabled, the SIR block uses the UnTx and UnRx pins for the SIR protocol. These signals must be connected to an infrared transceiver to implement an IrDA SIR physical layer link. The SIR block can receive and transmit, but the SIR block is only half-duplex so the SIR block cannot do both at the same time. Transmission must be stopped before data can be received. The IrDA SIR physical layer specifies a minimum 10ms delay between transmission and reception. The SIR block has two modes of operation:

- In normal IrDA mode, a zero logic level is transmitted as a high pulse of 3/16th duration of the selected baud rate bit period on the output pin, while logic one levels are transmitted as a static low signal. These levels control the driver of an infrared transmitter, sending a pulse of light for each zero. On the reception side, the incoming light pulses energize the photo transistor base of the receiver, pulling the output low and driving the UART input pin low.
- In low-power IrDA mode, the width of the transmitted infrared pulse is set to three times the period of the internally generated IrLPBaud16 signal (1.63µs, assuming a nominal 1.8432MHz frequency) by changing the appropriate bit in the UARTCTL register.

Whether the device is in normal or low-power IrDA mode, a start bit is deemed valid if the decoder is still low one period of IrLPBaud16 after the low was first detected. This enables a normal-mode UART to receive data from a low-power mode UART that can transmit pulses as small as 1.41µs. Thus, for both low-power and normal mode operation, the ILPDVSR field in the UARTILPR register must be programmed such that 1.42MHz < $f_{IrLPBaud16}$< 2.12MHz, resulting in a low-power pulse duration of 1.41 to 2.11µs (three times the period of IrLPBaud16). The minimum frequency of IrLPBaud16 make sure that pulses less than one period of IrLPBaud16 are rejected, but pulses greater than 1.4µs are accepted as valid pulses.

Figure 39-3 shows the UART transmit and receive signals, with and without IrDA modulation.

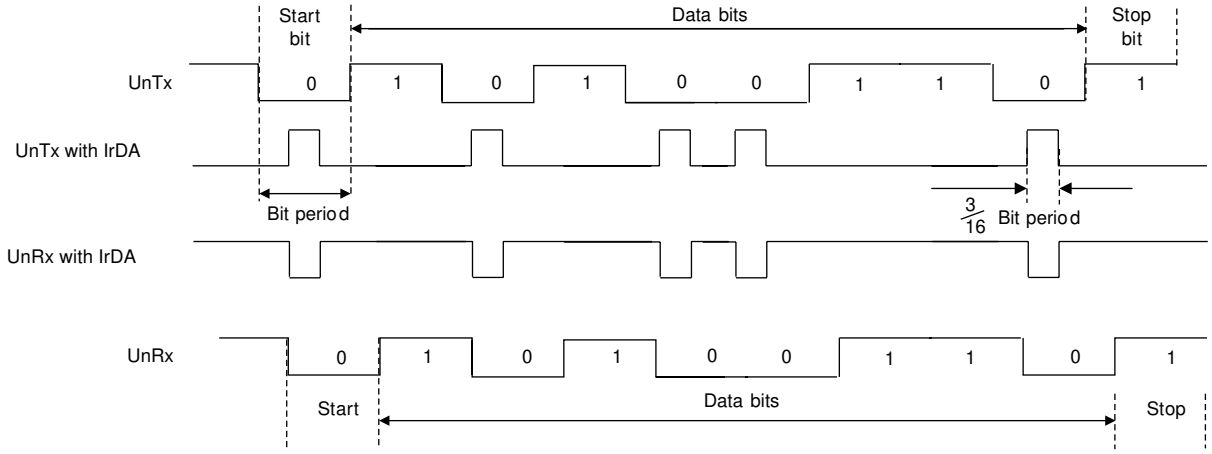


Figure 39-3. IrDA Data Modulation

In both normal and low-power IrDA modes:

- During transmission, the UART data bit is used as the base for encoding
- During reception, the decoded bits are transferred to the UART receive logic

The IrDA SIR physical layer specifies a half-duplex communication link, with a minimum 10ms delay between transmission and reception. This delay must be generated by software because the delay is not automatically supported by the UART. The delay is required because the infrared receiver electronics can become biased or even saturated from the optical power coupled from the adjacent transmitter LED. This delay is known as latency or receiver setup time.

39.2.5 9-Bit UART Mode

The UART provides a 9-bit mode that is enabled with the 9BITEN bit in the UART9BITADDR register. This feature is useful in a multi-drop configuration of the UART, where a single transmitter connected to multiple receivers can communicate with a particular receiver through the address or set of addresses along with a qualifier for an address byte. All the receivers check for the address qualifier in the place of the parity bit and, if set, then compare the byte received with the preprogrammed address. If the address matches, then the receiver receives or sends further data. If the address does not match, the receiver drops the address byte and any subsequent data bytes. If the UART is in 9-bit mode, then the receiver operates with no parity mode. The address can be predefined to match with the received byte and the address can be configured with the UART9BITADDR register. The matching can be extended to a set of addresses using the address mask in the UART9BITAMASK register. By default, the UART9BITAMASK is 0xFF, meaning that only the specified address is matched.

When not finding a match, the rest of the data bytes with the ninth bit cleared are dropped. If a match is found, then an interrupt is generated to the NVIC for further action. The subsequent data bytes with the cleared ninth bit are stored in the FIFO. Software can mask this interrupt in case RTDMA or FIFO operations are enabled for this instance and processor intervention is not required. All the send transactions with 9-bit mode are data bytes and the ninth bit is cleared. Software can override the ninth bit to be set (to indicate address) by overriding the parity settings to sticky parity with odd parity enabled for a particular byte. To match the transmission time with correct parity settings, the address byte can be transmitted as a single then a burst transfer. The transmit FIFO does not hold the address/data bit; hence, software can enable the address bit appropriately.

39.2.6 FIFO Operation

The UART has two 16-deep 8-bit wide FIFOs; one for transmit and one for receive. Both FIFOs are accessed by way of the UART Data (UARTDR) register. Read operations of the UARTDR register return a 12-bit value consisting of eight data bits and four error flags while write operations place 8-bit data in the transmit FIFO.

Out of reset, both FIFOs are disabled and act as 1 byte-deep holding registers. The FIFOs are enabled by setting the FEN bit in the UARTLCRH register.

FIFO status can be monitored through the UART Flag (UARTFR) register and the UART Receive Status (UARTRSR) register. Hardware monitors empty, full, and overrun conditions. The UARTFR register contains empty and full flags (TXFE, TXFF, RXFE, and RXFF bits), and the UARTRSR register shows overrun status with the OE bit. If the FIFOs are disabled, the empty and full flags are set according to the status of the 1 byte-deep holding registers.

The trigger points, at which the FIFOs generate interrupts, is controlled by way of the UART Interrupt FIFO Level Select (UARTIFLS) register. Both FIFOs can be individually configured to trigger interrupts at different levels. Available configurations include ⅛, ¼, ½, ¾, and ⅞ in the TXIFSEL and RXIFSEL fields of the UARTIFLS register.

For example, if the ¼ option is selected for the receive FIFO, the UART generates a receive interrupt after four data bytes are received. If the ¼ option is selected for the transmit FIFO, the UART generates a transmit interrupt after four data bytes can be transmitted, meaning there are at least four empty spaces in the transmit FIFO. Out of reset, both FIFOs are configured to trigger an interrupt at the ½ mark.

39.2.7 Interrupts

The UART can generate interrupts when the following conditions are observed:

- Overrun error
- Break error
- Parity error
- Framing error
- Receive time-out
- Transmit (when condition defined in the TXIFLSEL bit in the UARTIFLS register is met, or if the EOT bit in UARTCTL is set, when the last bit of all transmitted data leaves the serializer)
- Receive (when condition defined in the RXIFLSEL bit in the UARTIFLS register is met)

All of the interrupt events are ORed together before being sent to the interrupt controller, so the UART can only generate a single interrupt request to the controller at any given time. Software can service multiple interrupt events in a single interrupt service routine by reading the UART Masked Interrupt Status (UARTMIS) register.

The interrupt events that can trigger a controller-level interrupt are defined in the UART Interrupt Mask (UARTIM) register by setting the corresponding IM bits. If interrupts are not used, the raw interrupt status is visible by way of the UART Raw Interrupt Status (UARTRIS) register.

Note

For receive time-out, the RTIM bit in the UARTIM register must be set to see the RTMIS and RTRIS status in the UARTMIS and UARTRIS registers.

Interrupts are always cleared (for the UARTMIS and UARTRIS registers) by writing a 1 to the corresponding bit in the UART Interrupt Clear (UARTICR) register. Additionally, write a 1 to the INT_FLG_CLR field of the UART_GLB_INT_CLR register to clear the global interrupt flag and receive further UART interrupts.

The receive time-out interrupt is asserted when the receive FIFO is not empty, and no further data is received over a 32-bit period when the HSE bit is clear or over a 64-bit period when the HSE bit is set. The receive time-out interrupt is cleared either when the FIFO becomes empty through reading all the data (or by reading the holding register), or when a 1 is written to the corresponding bit in the UARTICR register.

The receive interrupt changes state when one of the following events occurs:

- If the FIFOs are enabled and the receive FIFO reaches the programmed trigger level, the RXRIS bit is set. The receive interrupt is cleared by reading data from the receive FIFO until the receive FIFO becomes less than the trigger level, or by clearing the interrupt by writing a 1 to the RXIC bit.
- If the FIFOs are disabled (have a depth of one location) and data is received thereby filling the location, the RXRIS bit is set. The receive interrupt is cleared by performing a single read of the receive FIFO, or by clearing the interrupt by writing a 1 to the RXIC bit.

The transmit interrupt changes state when one of the following events occurs:

- If the FIFOs are enabled and the transmit FIFO progresses through the programmed trigger level, the TXRIS bit is set. The transmit interrupt is based on a transition through level, therefore the FIFO must be written past the programmed trigger level otherwise no further transmit interrupts are generated. The transmit interrupt is cleared by writing data to the transmit FIFO until the transmit FIFO becomes greater than the trigger level, or by clearing the interrupt by writing a 1 to the TXIC bit.
- If the FIFOs are disabled (have a depth of one location) and there is no data present in the transmitters single location, the TXRIS bit is set. The transmit interrupt is cleared by performing a single write to the transmit FIFO, or by clearing the interrupt by writing a 1 to the TXIC bit.

39.2.8 Loopback Operation

The UART can be placed into an internal loopback mode for diagnostic or debug work by setting the LBE bit in the UARTCTL register. In loopback mode, data transmitted on the UnTx output is received on the UnRx input. Note that the LBE bit must be set before the UART is enabled.

39.2.9 RTDMA Operation

The UART provides an interface to the RTDMA controller with separate channels for transmit and receive. The RTDMA operation of the UART is enabled through the UART RTDMA Control (UARTDMACTL) register. When RTDMA operation is enabled, the UART asserts a RTDMA request on the receive or transmit channel when the associated FIFO can transfer data. To trigger the RTDMA with the UART trigger sources, the UART FIFOs on the corresponding channel must be enabled. RTDMA operation does not work in non-FIFO mode.

For the receive channel, a burst request is asserted whenever the amount of data in the receive FIFO is at or above the FIFO trigger level configured in the UARTIFLS register. For the transmit channel, a burst request is asserted whenever the transmit FIFO contains equal or fewer characters than the FIFO trigger level. The burst RTDMA transfer requests are handled automatically by the RTDMA controller depending on how the RTDMA channel is configured. When using the RTDMA to transfer 16-bit or 32-bit data to UARTDR for a transmit, only the 8 least-significant bits are transmitted.

To enable RTDMA operation for the receive channel, set the RXDMAE bit of the RTDMA Control (UARTDMACTL) register. To enable RTDMA operation for the transmit channel, set the TXDMAE bit of the UARTDMACTL register. The UART can also be configured to stop using RTDMA for the receive channel if a receive error occurs. If the DMAERR bit of the UARTDMACR register is set and a receive error occurs, the RTDMA receive requests are automatically disabled. This error condition can be cleared by clearing the appropriate UART error interrupt.

When the RTDMA controller is finished transferring data to the TX FIFO or from the RX FIFO, a dma_done signal is sent to the UART to indicate completion. The dma_done status is indicated through the DMATXRIS and DMARXIS bits of the UARTRIS register. An interrupt can be generated from these status bits by setting the DMATXIM and DMARXIM bits in the UARTIM register.

Note

The DMATXRIS bit can be used to indicate the completion of data transfer from the RTDMA controller to the TX FIFO. To indicate transfer completion from the serializer of the UART, the EOT bit can be enabled in the UARTCTL register.

See the *Real-Time Direct Memory Access (RTDMA)* chapter for more details about programming the RTDMA controller.

39.2.9.1 Receiving Data Using UART with RTDMA

When using the RTDMA with the RX FIFO, the RTDMA Burst Size [DMA_BURST_SIZE] must equal the number of bytes used to trigger the FIFO level interrupt defined by RXIFLSEL in UARTIFLS. For example, a level of 1/8 is triggered by 2 bytes or more present in the FIFO, so DMA_BURST_SIZE = 2. To make sure that the RTDMA correctly receives all data from the RX FIFO, the RTDMA Burst Size also must be an integer divisor of the total number of UART receives. For complete data reception, follow these steps:

1. Decide the total number of words to be received. [BUFFER_SIZE]
2. Decide the necessary FIFO level [UART_BUFFER_SIZE] and set RXIFSEL accordingly.
3. Calculate the number of RTDMA transfers. [DMA_TRANSFER_SIZE = BUFFER_SIZE / UART_BUFFER_SIZE]
4. Set the RTDMA Burst Size. [DMA_BURST_SIZE] equal to the UART_BUFFER_SIZE.
5. Configure RTDMA using the calculated values and use the UART peripheral trigger as the trigger source.

To receive 120 words from the UART using the RTDMA:

BUFFER_SIZE = 120

UART_BUFFER_SIZE = 12

RXIFSEL: UART_BUFFER_SIZE / 16 = 3/4

DMA_TRANSFER_SIZE: BUFFER_SIZE / UART_BUFFER-SIZE = 120/12 = 10

DMA_BURST_SIZE: UART_BUFFER_SIZE = 12

(1 burst = 12 words, 1 transfer = 10 bursts, 120 words received with each transfer)

39.2.9.2 Transmitting Data Using UART with RTDMA

When using the RTDMA with the TX FIFO, the RTDMA Burst Size [DMA_BURST_SIZE] must equal 16 - the number of bytes used to trigger the FIFO level interrupt defined by TXIFSEL in UARTIFLS. For example, a level of 1/8 is triggered by having 14 bytes or less present in the FIFO, so DMA_BURST_SIZE = 16 - 14 = 2. To make sure that the RTDMA writes all data correctly to the TX FIFO, the RTDMA Burst Size must also be an integer divisor of the total number of UART transmissions. For complete data transmission, follow these steps:

1. Decide the total number or words to be transmitted. [BUFFER_SIZE]
2. Decide the necessary FIFO level [UART_BUFFER_SIZE] and set TXIFSEL accordingly.
3. Calculate the number of RTDMA transfers. [DMA_TRANSFER_SIZE = BUFFER_SIZE / UART_BUFFER_SIZE]
4. Calculate the RTDMA Burst Size. [DMA_BURST_SIZE] to equal 16 - UART_BUFFER_SIZE.
5. Configure the RTDMA using the calculated values and use UART peripheral trigger as the trigger source.

Note

For a BUFFER_SIZE less than or equal to 16, a software trigger can alternatively be used.

To transmit 120 words from the UART using the RTDMA:

BUFFER_SIZE = 120

UART_BUFFER_SIZE = 12

TXIFSEL: 1 - (UART_BUFFER_SIZE / 16) = 1 - 3/4 = 1/4

DMA_TRANSFER_SIZE: (BUFFER_SIZE / UART_BUFFER-SIZE) = (120/12) = 10

DMA_BURST_SIZE: UART_BUFFER_SIZE = 12

(1 burst = 12 words, 1 transfer = 10 bursts, 120 words transmitted with each transfer)

39.3 Initialization and Configuration

To enable and initialize the UART, perform the following steps:

1. Disable the UART by clearing the UARTEN bit in the UARTCTL register.
2. Write the integer portion of the BRD to the UARTIBRD register.
3. Write the fractional portion of the BRD to the UARTFBRD register.
4. Write the desired serial parameters to the UARTLCRH register.
5. Optionally, configure the RTDMA channel (see the *Real-Time Direct Memory Access (RTDMA)* chapter and enable the RTDMA options in the UARTDMACTL register.
6. Enable the UART by setting the UARTEN bit in the UARTCTL register.

39.4 Software

39.4.1 UART Registers to Driverlib Functions

Table 39-1. UART Registers to Driverlib Functions

File	Driverlib Function
DR	
uart.c	UART_writeCharNonBlocking
uart.c	UART_writeCharArray
uart.c	UART_readCharArray
uart.c	UART_send9BitAddress
uart.h	UART_readCharNonBlocking
uart.h	UART_readChar
uart.h	UART_writeChar
RSR	
uart.h	UART_getRxError
FR	
uart.c	UART_writeCharNonBlocking
uart.c	UART_writeCharArray
uart.c	UART_readCharArray
uart.c	UART_send9BitAddress
uart.c	UART_stop9BitDataMode
uart.c	UART_configure9BitDataMode
uart.h	UART_disableModule
uart.h	UART_disableModuleNonFIFO
uart.h	UART_isDataAvailable
uart.h	UART_isSpaceAvailable
uart.h	UART_readCharNonBlocking
uart.h	UART_readChar
uart.h	UART_writeChar
uart.h	UART_isBusy
ILPR	
uart.h	UART_setIrDALPDivisor
IBRD	
uart.c	UART_setConfig
uart.c	UART_getConfig
FBRD	
uart.c	UART_setConfig
uart.c	UART_getConfig
LCRH	
uart.c	UART_setConfig
uart.c	UART_getConfig
uart.c	UART_send9BitAddress
uart.c	UART_stop9BitDataMode
uart.c	UART_configure9BitDataMode
uart.h	UART_setParityMode
uart.h	UART_getParityMode
uart.h	UART_enableModule

Table 39-1. UART Registers to Driverlib Functions (continued)

File	Driverlib Function
uart.h	UART_disableModule
uart.h	UART_enableFIFO
uart.h	UART_disableFIFO
uart.h	UART_isFIFOEnabled
uart.h	UART_setBreakConfig
CTL	
uart.c	UART_setConfig
uart.c	UART_getConfig
uart.h	UART_enableModule
uart.h	UART_enableModuleNonFIFO
uart.h	UART_disableModule
uart.h	UART_disableModuleNonFIFO
uart.h	UART_enableSIR
uart.h	UART_disableSIR
uart.h	UART_setTxIntMode
uart.h	UART_getTxIntMode
uart.h	UART_enableLoopback
uart.h	UART_disableLoopback
IFLS	
uart.h	UART_setFIFOLevel
uart.h	UART_getFIFOLevel
IM	
uart.h	UART_enableInterrupt
uart.h	UART_disableInterrupt
RIS	
uart.h	UART_getInterruptStatus
MIS	
uart.h	UART_getInterruptStatus
ICR	
uart.h	UART_clearInterruptStatus
DMACTL	
uart.h	UART_enableDMA
uart.h	UART_disableDMA
GLB_INT_FLG	
uart.h	UART_getGlobalInterruptFlagStatus
GLB_INT_CLR	
uart.h	UART_clearGlobalInterruptFlag
9BITADDR	
uart.h	UART_enable9Bit
uart.h	UART_disable9Bit
uart.h	UART_set9BitAddress
9BITAMASK	
uart.h	UART_set9BitAddress
PP	
-	

Table 39-1. UART Registers to Driverlib Functions (continued)

File	Driverlib Function
PERIPHID4	
-	
PERIPHID5	
-	
PERIPHID6	
-	
PERIPHID7	
-	
PERIPHID0	
-	
PERIPHID1	
-	
PERIPHID2	
-	
PERIPHID3	
-	
PCELLID0	
-	
PCELLID1	
-	
PCELLID2	
-	
PCELLID3	
-	
ECR	
uart.h	UART_clearRxError

39.4.2 UART Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location:
mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/uart

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

39.4.2.1 UART Loopback - SINGLE_CORE

FILE: uart_ex1_loopback.c

Simple UART internal loopback example

The sent data looks like this:
00 01 02 03 .... FE FF 00

This pattern is repeated forever.

External Connections

- None

Watch Variables

- *sData* - Data to send
- *rData* - Received data

Note: Avoid keeping the memory browser open while the execution is in progress.

39.4.2.2 UART Loopback with Interrupt - SINGLE_CORE

FILE: uart_ex2_loopback_fifo_interrupts.c

UART internal loopback w/ interrupt example Receive interrupt with FIFO is used.

A stream of data is sent and then compared to the received stream. The sent data looks like this:
00 01
01 02
02 03
....
FE FF
FF 00
etc..
The pattern is repeated forever.

External Connections
- None

Watch Variables
- *sData* - Data to send
- *rData* - Received data

Note: Avoid keeping the memory browser open while the execution is in progress.

39.4.2.3 UART Loopback with DMA - SINGLE_CORE

FILE: uart_ex3_loopback_dma.c

This program uses the internal loopback test mode of the UART module. Both DMA interrupts and UART FIFOs are used. When the UART transmit FIFO has enough space (as indicated by its FIFO level interrupt signal), the DMA will transfer data from global variable sData into the FIFO. This will be transmitted to the receive FIFO via the internal loopback.

When enough data has been placed in the receive FIFO (as indicated by its FIFO level interrupt signal), the DMA will transfer the data from the FIFO into global variable rData.

When all data has been placed into rData, a check of the validity of the data will be performed in one of the DMA channels' ISRs.

External Connections
- None

Watch Variables
- *sData* - Data to send
- *rData* - Received data

39.4.2.4 UART Echoback - SINGLE_CORE

FILE: uart_ex4_echoback.c

This test receives and echo-backs data through the UART A port.

A terminal such as 'putty' can be used to view the data from the UART and to send information to the UART. Characters received by the UART port are sent back to the host.

Running the Application Open a COM port with the following settings using a terminal:
- Find correct COM port
- Bits per second = 115200
- Data Bits = 8

- Parity = None
- Stop Bits = 1
- Hardware Control = None

The program will print out a greeting and then ask you to enter a character which it will echo back to the terminal.

Watch Variables

- None

External Connections

Connect the UARTA port to a PC via a transceiver and cable.

- GPIO43 is UARTA RX
- GPIO42 is UARTA TX

Note: Avoid keeping the memory browser open while the execution is in progress.

39.5 UART Registers

This Section describes the UART Registers.

39.5.1 UART Base Address Table

Table 39-2. UART Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
UART_REGS , UART_REGS_WRITE	UARTA_BASE, UARTA_WRITE_BASE	0x6007_0000	YES	YES	YES	YES	YES	YES	-	YES
UART_REGS , UART_REGS_WRITE	UARTB_BASE, UARTB_WRITE_BASE	0x6007_2000	YES	YES	YES	YES	YES	YES	-	YES
UART_REGS , UART_REGS_WRITE	UARTC_BASE, UARTC_WRITE_BASE	0x6007_4000	YES	YES	YES	YES	YES	YES	-	YES
UART_REGS , UART_REGS_WRITE	UARTD_BASE, UARTD_WRITE_BASE	0x6007_6000	YES	YES	YES	YES	YES	YES	-	YES
UART_REGS , UART_REGS_WRITE	UARTE_BASE, UARTE_WRITE_BASE	0x6007_8000	YES	YES	YES	YES	YES	YES	-	YES
UART_REGS , UART_REGS_WRITE	UARTF_BASE, UARTF_WRITE_BASE	0x6007_A000	YES	YES	YES	YES	YES	YES	-	YES

39.5.2 UART_REGS Registers

Table 39-3 lists the memory-mapped registers for the UART_REGS registers. All register offset addresses not listed in Table 39-3 should be considered as reserved locations and the register contents should not be modified.

Table 39-3. UART_REGS Registers

Offset	Acronym	Register Name	Protection
0h	UARTDR	UART Data	
4h	UARTRSR	UART Receive Status/Error Clear	
18h	UARTFR	UART Flag	
20h	UARTILPR	UART IrDA Low-Power Register	
24h	UARTIBRD	UART Integer Baud-Rate Divisor	
28h	UARTFBRD	UART Fractional Baud-Rate Divisor	
2Ch	UARTLCRH	UART Line Control	
30h	UARTCTL	UART Control	
34h	UARTIFLS	UART Interrupt FIFO Level Select	
38h	UARTIM	UART Interrupt Mask	
3Ch	UARTRIS	UART Raw Interrupt Status	
40h	UARTMIS	UART Masked Interrupt Status	
44h	UARTICR	UART Interrupt Clear	
48h	UARTDMACTL	UART DMA Control	
80h	UART_GLB_INT_EN	UART Global Interrupt Enable Register	
84h	UART_GLB_INT_FLG	UART Global Interrupt Flag Register	
88h	UART_GLB_INT_CLR	UART Global Interrupt Clear Register	
A4h	UART9BITADDR	UART 9-Bit Self Address	
A8h	UART9BITAMASK	UART 9-Bit Self Address Mask	
FC0h	UARTPP	UART Peripheral Properties	
FD0h	UARTPeriphID4	UART Peripheral Identification 4	
FD4h	UARTPeriphID5	UART Peripheral Identification 5	
FD8h	UARTPeriphID6	UART Peripheral Identification 6	
FDCh	UARTPeriphID7	UART Peripheral Identification 7	
FE0h	UARTPeriphID0	UART Peripheral Identification 0	
FE4h	UARTPeriphID1	UART Peripheral Identification 1	
FE8h	UARTPeriphID2	UART Peripheral Identification 2	
FECh	UARTPeriphID3	UART Peripheral Identification 3	
FF0h	UARTPCellID0	UART PrimeCell Identification 0	
FF4h	UARTPCellID1	UART PrimeCell Identification 1	
FF8h	UARTPCellID2	UART PrimeCell Identification 2	
FFCh	UARTPCellID3	UART PrimeCell Identification 3	

Complex bit access types are encoded to fit into small table cells. Table 39-4 shows the codes that are used for access types in this section.

Table 39-4. UART_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		

Table 39-4. UART_REGS Access Type Codes (continued)

Access Type	Code	Description
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

39.5.2.1 UARTDR Register (Offset = 0h) [Reset = 00000000h]

UARTDR is shown in Figure 39-4 and described in Table 39-5.

Return to the Summary Table.

IMPORTANT: This register is read sensitive. This register is the data register (the interface to the FIFOs). For transmitted data, if the FIFO is enabled, data written to this location is pushed onto the transmit FIFO. If the FIFO is disabled, data is stored in the transmitter holding register (the bottom word of the transmit FIFO). A write to this register initiates a transmission from the UART.
For received data, if the FIFO is enabled, the data byte and the 4-bit status (break, frame, parity, and overrun) is pushed onto the 12-bit wide receive FIFO. If the FIFO is disabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO). The received data can be retrieved by reading this register.

Figure 39-4. UARTDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				OE	BE	PE	FE	DATA							
R-0h				R-0h	R-0h	R-0h	R-0h	R/W-0h							

Table 39-5. UARTDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11	OE	R	0h	UART Overrun Error 0 No data has been lost due to a FIFO overrun. 1 New data was received when the FIFO was full, resulting in data loss. Reset type: PER.RESET
10	BE	R	0h	UART Break Error 0 No break condition has occurred 1 A break condition has been detected, indicating that the receive data input was held Low for longer than a full-word transmission time (defined as start, data, parity, and stop bits). In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the received data input goes to a 1 (marking state), and the next valid start bit is received. Reset type: PER.RESET
9	PE	R	0h	UART Parity Error 0 No parity error has occurred 1 The parity of the received data character does not match the parity defined by bits 2 and 7 of the UARTLCRH register. In FIFO mode, this error is associated with the character at the top of the FIFO. Reset type: PER.RESET
8	FE	R	0h	UART Framing Error 0 No framing error has occurred 1 The received character does not have a valid stop bit (a valid stop bit is 1). Reset type: PER.RESET

Table 39-5. UARTDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	DATA	R/W	0h	Data Transmitted or Received Data that is to be transmitted via the UART is written to this field. When read, this field contains the data that was received by the UART. For transmitted data, if the FIFO is enabled, data written to this location is pushed onto the transmit FIFO. If the FIFO is disabled, data is stored in the transmitter holding register (the bottom word of the transmit FIFO). A write to this register initiates a transmission from the UART. For received data, if the FIFO is enabled, the data byte and the 4-bit status (break, frame, parity, and overrun) is pushed onto the 12-bit wide receive FIFO. If the FIFO is disabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO). The received data can be retrieved by reading this register. Reset type: PER.RESET

39.5.2.2 UARTRSR Register (Offset = 4h) [Reset = 00000000h]

UARTRSR is shown in Figure 39-5 and described in Table 39-6.

Return to the Summary Table.

The UARTRSR/UARTECR register is the receive status register/error clear register.
In addition to the UARTDR register, receive status can also be read from the UARTRSR register.
If the status is read from this register, then the status information corresponds to the entry read from UARTDR prior to reading UARTRSR. The status information for overrun is set immediately when an overrun condition occurs.
The UARTRSR register cannot be written.
A write of any value to the UARTECR register clears the framing, parity, break, and overrun errors.
All the bits are cleared on reset.

Figure 39-5. UARTRSR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												OE	BE	PE	FE
R-0h												R-0h	R-0h	R-0h	R-0h

Table 39-6. UARTRSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3	OE	R	0h	UART Overrun Error 0 No data has been lost due to a FIFO overrun. 1 New data was received when the FIFO was full, resulting in data loss. This bit is cleared by a write to UARTECR.The FIFO contents remain valid because no further data is written when the FIFO is full, only the contents of the shift register are overwritten. The CPU must read the data in order to empty the FIFO. Reset type: PER.RESET
2	BE	R	0h	UART Break Error 0 No break condition has occurred 1 A break condition has been detected, indicating that the receive data input was held Low for longer than a full-word transmission time (defined as start, data, parity, and stop bits). This bit is cleared to 0 by a write to UARTECR. In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to a 1 (marking state) and the next valid start bit is received. Reset type: PER.RESET
1	PE	R	0h	UART Parity Error 0 No parity error has occurred 1 The parity of the received data character does not match the parity defined by bits 2 and 7 of the UARTLCRH register. This bit is cleared to 0 by a write to UARTECR. Reset type: PER.RESET

Table 39-6. UARTRSR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	FE	R	0h	UART Framing Error 0 No framing error has occurred 1 The received character does not have a valid stop bit (a valid stop bit is 1). This bit is cleared to 0 by a write to UARTECR. In FIFO mode, this error is associated with the character at the top of the FIFO. Reset type: PER.RESET

39.5.2.3 UARTFR Register (Offset = 18h) [Reset = 00000090h]

UARTFR is shown in Figure 39-6 and described in Table 39-7.

Return to the Summary Table.

The UARTFR register is the flag register. After reset, the TXFF, RXFF, and BUSY bits are 0, and TXFE and RXFE bits are 1.

Figure 39-6. UARTFR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							RESERVED
R-0h							R-0h
7	6	5	4	3	2	1	0
TXFE	RXFF	TXFF	RXFE	BUSY	RESERVED	RESERVED	RESERVED
R-1h	R-0h	R-0h	R-1h	R-0h	R-0h	R-0h	R-0h

Table 39-7. UARTFR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7	TXFE	R	1h	UART Transmit FIFO Empty The meaning of this bit depends on the state of the FEN bit in the UARTLCRH register. 0 The transmitter has data to transmit. 1 If the FIFO is disabled (FEN is 0), the transmit holding register is empty. If the FIFO is enabled (FEN is 1), the transmit FIFO is empty. Reset type: PER.RESET
6	RXFF	R	0h	UART Receive FIFO Full The meaning of this bit depends on the state of the FEN bit in the UARTLCRH register. 0 The receiver can receive data. 1 If the FIFO is disabled (FEN is 0), the receive holding register is full. If the FIFO is enabled (FEN is 1), the receive FIFO is full. Reset type: PER.RESET
5	TXFF	R	0h	UART Transmit FIFO Full The meaning of this bit depends on the state of the FEN bit in the UARTLCRH register. 0 The transmitter is not full. 1 If the FIFO is disabled (FEN is 0), the transmit holding register is full. If the FIFO is enabled (FEN is 1), the transmit FIFO is full. Reset type: PER.RESET

Table 39-7. UARTFR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	RXFE	R	1h	UART Receive FIFO Empty The meaning of this bit depends on the state of the FEN bit in the UARTLCRH register. 0 The receiver is not empty. 1 If the FIFO is disabled (FEN is 0), the receive holding register is empty. If the FIFO is enabled (FEN is 1), the receive FIFO is empty. Reset type: PER.RESET
3	BUSY	R	0h	UART Busy 0 The UART is not busy. 1 The UART is busy transmitting data. This bit remains set until the complete byte, including all stop bits, has been sent from the shift register. This bit is set as soon as the transmit FIFO becomes non-empty (regardless of whether UART is enabled). Reset type: PER.RESET
2	RESERVED	R	0h	Reserved
1	RESERVED	R	0h	Reserved
0	RESERVED	R	0h	Reserved

39.5.2.4 UARTILPR Register (Offset = 20h) [Reset = 00000000h]

UARTILPR is shown in Figure 39-7 and described in Table 39-8.

Return to the Summary Table.

The UARTILPR register stores the 8-bit low-power counter divisor value used to derive the low-power SIR pulse width clock.

Figure 39-7. UARTILPR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ILPDVSR							
R-0h																								R/W-0h							

Table 39-8. UARTILPR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	ILPDVSR	R/W	0h	IrDA Low-Power Divisor This field contains the 8-bit low-power divisor value. The UARTILPR register stores the 8-bit low-power counter divisor value used to derive the low-power SIR pulse width clock by dividing down the system clock (SysClk). All the bits are cleared when reset. The internal IrLPBaud16 clock is generated by dividing down SysClk according to the low-power divisor value written to UARTILPR. The duration of SIR pulses generated when low-power mode is enabled is three times the period of the IrLPBaud16 clock. The low-power divisor value is calculated as follows: ILPDVSR = SysClk / FIrLPBaud16 where FIrLPBaud16 is nominally 1.8432 MHz. Because the IrLPBaud16 clock is used to sample transmitted data irrespective of mode, the ILPDVSR field must be programmed in both low power and normal mode,such that FIrLPBaud16 is between 1.42 and 2.12 MHz, resulting in a low-power pulse duration of 1.41-2.11 us (three times the period of IrLPBaud16). The minimum frequency of IrLPBaud16 ensures that pulses less than one period of IrLPBaud16 are rejected, but pulses greater than 1.4 us are accepted as valid pulses. Note: Zero is an illegal value. Programming a zero value results in no IrLPBaud16 pulses being generated Reset type: PER.RESET

39.5.2.5 UARTIBRD Register (Offset = 24h) [Reset = 00000000h]

UARTIBRD is shown in Figure 39-8 and described in Table 39-9.

Return to the Summary Table.

The UARTIBRD register is the integer part of the baud-rate divisor value. All the bits are cleared on reset. The minimum possible divide ratio is 1 (when UARTIBRD=0), in which case the UARTFBRD register is ignored. When changing the UARTIBRD register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register.

Figure 39-8. UARTIBRD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DIVINT															
R-0h																R/W-0h															

Table 39-9. UARTIBRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	DIVINT	R/W	0h	Integer Baud-Rate Divisor The minimum possible divide ratio is 1 (when UARTIBRD=0), in which case the UARTFBRD register is ignored. When changing the UARTIBRD register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register. Reset type: PER.RESET

39.5.2.6 UARTFBRD Register (Offset = 28h) [Reset = 00000000h]

UARTFBRD is shown in Figure 39-9 and described in Table 39-10.

Return to the Summary Table.

The UARTFBRD register is the fractional part of the baud-rate divisor value. All the bits are cleared on reset. When changing the UARTFBRD register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register. See 'Baud-Rate Generation' on page 1165 for configuration details.

Figure 39-9. UARTFBRD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										DIVFRAC					
R-0h										R/W-0h					

Table 39-10. UARTFBRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-0	DIVFRAC	R/W	0h	Fractional Baud-Rate Divisor The UARTFBRD register is the fractional part of the baud-rate divisor value. All the bits are cleared on reset. When changing the UARTFBRD register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register. Reset type: PER.RESET

39.5.2.7 UARTLCRH Register (Offset = 2Ch) [Reset = 00000000h]

UARTLCRH is shown in Figure 39-10 and described in Table 39-11.

Return to the Summary Table.

The UARTLCRH register is the line control register. Serial parameters such as data length, parity, and stop bit selection are implemented in this register.
When updating the baud-rate divisor (UARTIBRD and/or UARTIFRD), the UARTLCRH register must also be written. The write strobe for the baud-rate divisor registers is tied to the UARTLCRH register.

Figure 39-10. UARTLCRH Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
SPS	WLEN		FEN	STP2	EPS	PEN	BRK
R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 39-11. UARTLCRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	SPS	R/W	0h	UART Stick Parity Select UART Stick Parity Select 0 Stick parity is disabled (default) 1 Stick parity is enabled. When bits 1, 2, and 7 of UARTLCRH are set, the parity bit is transmitted and checked as a 0. When bits 1 and 7 are set and 2 is cleared, the parity bit is transmitted and checked as a 1. Reset type: PER.RESET
6-5	WLEN	R/W	0h	UART Word Length The bits indicate the number of data bits transmitted or received in a frame as follows: 0x0 5 bits (default) 0x1 6 bits 0x2 7 bits 0x3 8 bits Reset type: PER.RESET
4	FEN	R/W	0h	UART Enable FIFOs 0 The FIFOs are disabled. The FIFOs become 1-byte-deep holding registers. 1 The transmit and receive FIFO buffers are enabled (FIFO mode). Reset type: PER.RESET
3	STP2	R/W	0h	UART Two Stop Bits Select 0 One stop bit is transmitted at the end of a frame. 1 Two stop bits are transmitted at the end of a frame. The receive logic does not check for two stop bits being received. Reset type: PER.RESET

Table 39-11. UARTLCRH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	EPS	R/W	0h	UART Even Parity Select 0 Odd parity is performed, which checks for an odd number of 1s. 1 Even parity generation and checking is performed during transmission and reception, which checks for an even number of 1s in data and parity bits. This bit has no effect when parity is disabled by the PEN bit. Reset type: PER.RESET
1	PEN	R/W	0h	UART Parity Enable 0 Parity is disabled and no parity bit is added to the data frame. 1 Parity checking and generation is enabled. Reset type: PER.RESET
0	BRK	R/W	0h	UART Send Break 0 Normal use. 1 A Low level is continually output on the UnTx signal, after completing transmission of the current character. For the proper execution of the break command, software must set this bit for at least two frames (character periods). Reset type: PER.RESET

39.5.2.8 UARTCTL Register (Offset = 30h) [Reset = 00000300h]

UARTCTL is shown in Figure 39-11 and described in Table 39-12.

Return to the Summary Table.

The UARTCTL register is the control register. All the bits are cleared on reset except for the Transmit Enable (TXE) and Receive Enable (RXE) bits, which are set.
To enable the UART module, the UARTEN bit must be set. If software requires a configuration change in the module, the UARTEN bit must be cleared before the configuration changes are written. If the UART is disabled during a transmit or receive operation, the current transaction is completed prior to the UART stopping.

Figure 39-11. UARTCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED		RESERVED	RESERVED	RXE	TXE
R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
LBE	RESERVED	HSE	EOT	RESERVED	SIRLP	SIREN	UARTEN
R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 39-12. UARTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	RXE	R/W	1h	UART Receive Enable 0 The receive section of the UART is disabled. 1 The receive section of the UART is enabled. If the UART is disabled in the middle of a receive, it completes the current character before stopping.To enable reception, the UARTEN bit must also be set. Reset type: PER.RESET
8	TXE	R/W	1h	UART Transmit Enable 0 The transmit section of the UART is disabled. 1 The transmit section of the UART is enabled. If the UART is disabled in the middle of a transmission, it completes the current character before stopping. To enable transmission, the UARTEN bit must also be set. Reset type: PER.RESET
7	LBE	R/W	0h	UART Loop Back Enable 0 Normal operation. 1 The UnTx path is fed through the UnRx path. Reset type: PER.RESET
6	RESERVED	R	0h	Reserved

Table 39-12. UARTCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	HSE	R/W	0h	High-Speed Enable 0 The UART is clocked using the system clock divided by 16. 1 The UART is clocked using the system clock divided by 8. Reset type: PER.RESET
4	EOT	R/W	0h	End of Transmission This bit determines the behavior of the TXRIS bit in the UARTRIS register. 0 The TXRIS bit is set when the transmit FIFO condition specified in UARTIFLS is met. 1 The TXRIS bit is set only after all transmitted data, including stop bits, have cleared the serializer. Reset type: PER.RESET
3	RESERVED	R/W	0h	Reserved
2	SIRLP	R/W	0h	UART SIR Low-Power Mode This bit selects the IrDA encoding mode. 0 Low-level bits are transmitted as an active High pulse with a width of 3/16th of the bit period. 1 The UART operates in SIR Low-Power mode. Low-level bits are transmitted with a pulse width which is 3 times the period of the IrLPBaud16 input signal, regardless of the selected bit rate. Setting this bit uses less power, but might reduce transmission distances. Reset type: PER.RESET
1	SIREN	R/W	0h	UART SIR Enable 0 Normal operation. 1 The IrDA SIR block is enabled, and the UART will transmit and receive data using SIR protocol. Reset type: PER.RESET
0	UARTEN	R/W	0h	UART Enable 0 The UART is disabled. 1 The UART is enabled. If the UART is disabled in the middle of transmission or reception, it completes the current character before stopping. Reset type: PER.RESET

39.5.2.9 UARTIFLS Register (Offset = 34h) [Reset = 00000012h]

UARTIFLS is shown in Figure 39-12 and described in Table 39-13.

Return to the Summary Table.

The UARTIFLS register is the interrupt FIFO level select register. You can use this register to define the FIFO level at which the TXRIS and RXRIS bits in the UARTRIS register are triggered.
The interrupts are generated based on a transition through a level rather than being based on the level. That is, the interrupts are generated when the fill level progresses through the trigger level.
For example, if the receive trigger level is set to the half-way mark, the interrupt is triggered as the module is receiving the 9th character.
Therefore, changing the trigger level does not trigger an interrupt using the new level until another character is received.
Out of reset, the TXIFLSEL and RXIFLSEL bits are configured so that the FIFOs trigger an interrupt at the half-way mark.

Figure 39-12. UARTIFLS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										RXIFLSEL			TXIFLSEL		
R-0h										R/W-2h			R/W-2h		

Table 39-13. UARTIFLS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-3	RXIFLSEL	R/W	2h	UART Receive Interrupt FIFO Level Select The trigger points for the receive interrupt are as follows: Value Description 0x0 RX FIFO greater than or equal to 1/8 full - at least 2 filled spots 0x1 RX FIFO greater than or equal to 1/4 full - at least 4 filled spots 0x2 RX FIFO greater than or equal to 1/2 full - at least 8 filled spots (default) 0x3 RX FIFO greater than or equal to 3/4 full - at least 12 filled spots 0x4 RX FIFO greater than or equal to 7/8 full - at least 14 filled spots 0x5-0x7 Reserved Reset type: PER.RESET
2-0	TXIFLSEL	R/W	2h	Value Description 0x0 TX FIFO less than or equal to 1/8 full - at least 14 empty spots 0x1 TX FIFO less than or equal to 1/4 full - at least 12 empty spots 0x2 TX FIFO less than or equal to 1/2 full - at least 8 empty spots (default) 0x3 TX FIFO less than or equal to 3/4 full - at least 4 empty spots 0x4 TX FIFO less than or equal to 7/8 full - at least 2 empty spots 0x5-0x7 Reserved Note: If the EOT bit in UARTCTL is set, the transmit interrupt is generated once the FIFO is completely empty and all data including stop bits have left the transmit serializer. In this case, the setting of TXIFLSEL is ignored. Reset type: PER.RESET

39.5.2.10 UARTIM Register (Offset = 38h) [Reset = 00000000h]

UARTIM is shown in Figure 39-13 and described in Table 39-14.

Return to the Summary Table.

The UARTIM register is the interrupt mask set/clear register.
On a read, this register gives the current value of the mask on the relevant interrupt. Setting a bit allows the corresponding raw interrupt signal to be routed to the interrupt controller. Clearing a bit prevents the raw interrupt signal from being sent to the interrupt controller.

Figure 39-13. UARTIM Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						DMATXIM	DMARXIM
R-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED			9BITIM	RESERVED	OEIM	BEIM	PEIM
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
FEIM	RTIM	TXIM	RXIM	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 39-14. UARTIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17	DMATXIM	R/W	0h	Transmit DMA Interrupt Mask 0 The DMATXRIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the DMATXRIS bit in the UARTRIS register is set. Reset type: PER.RESET
16	DMARXIM	R/W	0h	Receive DMA Interrupt Mask 0 The DMARXRIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the DMARXRIS bit in the UARTRIS register is set. Reset type: PER.RESET
15-13	RESERVED	R	0h	Reserved
12	9BITIM	R/W	0h	9-Bit Mode Interrupt Mask 0 The 9BITRIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the 9BITRIS bit in the UARTRIS register is set. Reset type: PER.RESET
11	RESERVED	R/W	0h	Reserved
10	OEIM	R/W	0h	UART Overrun Error Interrupt Mask 0 The OERIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the OERIS bit in the UARTRIS register is set. Reset type: PER.RESET

Table 39-14. UARTIM Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	BEIM	R/W	0h	UART Break Error Interrupt Mask 0 The BERIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the BERIS bit in the UARTRIS register is set. Reset type: PER.RESET
8	PEIM	R/W	0h	UART Parity Error Interrupt Mask 0 The PERIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the PERIS bit in the UARTRIS register is set. Reset type: PER.RESET
7	FEIM	R/W	0h	UART Framing Error Interrupt Mask 0 The FERIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the FERIS bit in the UARTRIS register is set. Reset type: PER.RESET
6	RTIM	R/W	0h	UART Receive Time-Out Interrupt Mask 0 The RTRIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the RTRIS bit in the UARTRIS register is set. Reset type: PER.RESET
5	TXIM	R/W	0h	UART Transmit Interrupt Mask 0 The TXRIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the TXRIS bit in the UARTRIS register is set. Reset type: PER.RESET
4	RXIM	R/W	0h	UART Receive Interrupt Mask 0 The RXRIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the RXRIS bit in the UARTRIS register is set. Reset type: PER.RESET
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

39.5.2.11 UARTRIS Register (Offset = 3Ch) [Reset = 00000000h]

UARTRIS is shown in Figure 39-14 and described in Table 39-15.

Return to the Summary Table.

The UARTRIS register is the raw interrupt status register. On a read, this register gives the current raw status value of the corresponding interrupt. A write has no effect.

Figure 39-14. UARTRIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						DMATXRIS	DMARXRIS
R-0h						R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED			9BITRIS	RESERVED	OERIS	BERIS	PERIS
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
FERIS	RTRIS	TXRIS	RXRIS	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 39-15. UARTRIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17	DMATXRIS	R	0h	Transmit DMA Raw Interrupt Status 0 No interrupt 1 The transmit DMA has completed. This bit is cleared by writing a 1 to the DMATXIC bit in the UARTICR register. Reset type: PER.RESET
16	DMARXRIS	R	0h	Receive DMA Raw Interrupt Status 0 No interrupt 1 The receive DMA has completed. This bit is cleared by writing a 1 to the DMARXIC bit in the UARTICR register. Reset type: PER.RESET
15-13	RESERVED	R	0h	Reserved
12	9BITRIS	R	0h	9-Bit Mode Raw Interrupt Status 0 No interrupt 1 A receive address match has occurred. This bit is cleared by writing a 1 to the 9BITIC bit in the UARTICR register. Reset type: PER.RESET
11	RESERVED	R	0h	Reserved
10	OERIS	R	0h	UART Overrun Error Raw Interrupt Status 0 No interrupt 1 An overrun error has occurred. This bit is cleared by writing a 1 to the OEIC bit in the UARTICR register. Reset type: PER.RESET

Table 39-15. UARTRIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	BERIS	R	0h	UART Break Error Raw Interrupt Status 0 No interrupt 1 A break error has occurred. This bit is cleared by writing a 1 to the BEIC bit in the UARTICR register. Reset type: PER.RESET
8	PERIS	R	0h	UART Parity Error Raw Interrupt Status 0 No interrupt 1 A parity error has occurred. This bit is cleared by writing a 1 to the PEIC bit in the UARTICR register. Reset type: PER.RESET
7	FERIS	R	0h	UART Framing Error Raw Interrupt Status 0 No interrupt 1 A framing error has occurred. This bit is cleared by writing a 1 to the FEIC bit in the UARTICR register. Reset type: PER.RESET
6	RTRIS	R	0h	UART Receive Time-Out Raw Interrupt Status 0 No interrupt 1 A receive time out has occurred. This bit is cleared by writing a 1 to the RTIC bit in the UARTICR register. Reset type: PER.RESET
5	TXRIS	R	0h	UART Transmit Raw Interrupt Status 0 No interrupt 1 If the EOT bit in the UARTCTL register is clear, the transmit FIFO level has passed through the condition defined in the UARTIFLS register.If the EOT bit is set, the last bit of all transmitted data and flags has left the serializer. This bit is cleared by writing a 1 to the TXIC bit in the UARTICR register or by writing data to the transmit FIFO until it becomes greater than the trigger level, if the FIFO is enabled, or by writing a single byte if the FIFO is disabled. Reset type: PER.RESET
4	RXRIS	R	0h	UART Receive Raw Interrupt Status 0 No interrupt 1 The receive FIFO level has passed through the condition defined in the UARTIFLS register. This bit is cleared by writing a 1 to the RXIC bit in the UARTICR register or by reading data from the receive FIFO until it becomes less than the trigger level, if the FIFO is enabled, or by reading a single byte if the FIFO is disabled. Reset type: PER.RESET
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	RESERVED	R	0h	Reserved
0	RESERVED	R	0h	Reserved

39.5.2.12 UARTMIS Register (Offset = 40h) [Reset = 00000000h]

UARTMIS is shown in Figure 39-15 and described in Table 39-16.

Return to the Summary Table.

The UARTMIS register is the masked interrupt status register. On a read, this register gives the current masked status value of the corresponding interrupt. A write has no effect.

Figure 39-15. UARTMIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						DMATXMIS	DMARXMIS
R-0h						R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED			9BITMIS	RESERVED	OEMIS	BEMIS	PEMIS
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
FEMIS	RTMIS	TXMIS	RXMIS	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 39-16. UARTMIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17	DMATXMIS	R	0h	Transmit DMA Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to the completion of the transmit DMA. This bit is cleared by writing a 1 to the DMATXIC bit in the UARTICR register. Reset type: PER.RESET
16	DMARXMIS	R	0h	Receive DMA Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to the completion of the receive DMA. This bit is cleared by writing a 1 to the DMARXIC bit in the UARTICR register. Reset type: PER.RESET
15-13	RESERVED	R	0h	Reserved
12	9BITMIS	R	0h	9-Bit Mode Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to a receive address match. This bit is cleared by writing a 1 to the 9BITIC bit in the UARTICR register. Reset type: PER.RESET
11	RESERVED	R	0h	Reserved
10	OEMIS	R	0h	UART Overrun Error Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to an overrun error. This bit is cleared by writing a 1 to the OEIC bit in the UARTICR register. Reset type: PER.RESET

Table 39-16. UARTMIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	BEMIS	R	0h	UART Break Error Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to a break error. This bit is cleared by writing a 1 to the BEIC bit in the UARTICR register. Reset type: PER.RESET
8	PEMIS	R	0h	UART Parity Error Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to a parity error. This bit is cleared by writing a 1 to the PEIC bit in the UARTICR register. Reset type: PER.RESET
7	FEMIS	R	0h	UART Framing Error Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to a framing error. This bit is cleared by writing a 1 to the FEIC bit in the UARTICR register. Reset type: PER.RESET
6	RTMIS	R	0h	UART Receive Time-Out Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to a receive time out. This bit is cleared by writing a 1 to the RTIC bit in the UARTICR register. Reset type: PER.RESET
5	TXMIS	R	0h	UART Transmit Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to passing through the specified transmit FIFO level (if the EOT bit is clear) or due to the transmission of the last data bit (if the EOT bit is set). This bit is cleared by writing a 1 to the TXIC bit in the UARTICR register or by writing data to the transmit FIFO until it becomes greater than the trigger level, if the FIFO is enabled, or by writing a single byte if the FIFO is disabled. Reset type: PER.RESET
4	RXMIS	R	0h	UART Receive Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to passing through the specified receive FIFO level. This bit is cleared by writing a 1 to the RXIC bit in the UARTICR register or by reading data from the receive FIFO until it becomes less than the trigger level, if the FIFO is enabled, or by reading a single byte if the FIFO is disabled. Reset type: PER.RESET
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	RESERVED	R	0h	Reserved
0	RESERVED	R	0h	Reserved

39.5.2.13 UARTICR Register (Offset = 44h) [Reset = 00000000h]

UARTICR is shown in Figure 39-16 and described in Table 39-17.

Return to the Summary Table.

The UARTICR register is the interrupt clear register. On a write of 1, the corresponding interrupt (both raw interrupt and masked interrupt, if enabled) is cleared. A write of 0 has no effect.

Figure 39-16. UARTICR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						DMATXIC	DMARXIC
R-0h						R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED			9BITIC	EOTIC	OEIC	BEIC	PEIC
R-0h			R/W-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
FEIC	RTIC	TXIC	RXIC	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 39-17. UARTICR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17	DMATXIC	R-0/W1S	0h	Transmit DMA Interrupt Clear Writing a 1 to this bit clears the DMATXRIS bit in the UARTRIS register and the DMATXMIS bit in the UARTMIS register. Reset type: PER.RESET
16	DMARXIC	R-0/W1S	0h	Receive DMA Interrupt Clear Writing a 1 to this bit clears the DMARXRIS bit in the UARTRIS register and the DMARXMIS bit in the UARTMIS register. Reset type: PER.RESET
15-13	RESERVED	R	0h	Reserved
12	9BITIC	R/W	0h	9-Bit Mode Interrupt Clear Writing a 1 to this bit clears the 9BITRIS bit in the UARTRIS register and the 9BITMIS bit in the UARTMIS register. Reset type: PER.RESET
11	EOTIC	R-0/W1S	0h	End of Transmission Interrupt Clear Writing a 1 to this bit clears the EOTRIS bit in the UARTRIS register and the EOTMIS bit in the UARTMIS register. Reset type: PER.RESET
10	OEIC	R-0/W1S	0h	Overrun Error Interrupt Clear Writing a 1 to this bit clears the OERIS bit in the UARTRIS register and the OEMIS bit in the UARTMIS register. Reset type: PER.RESET
9	BEIC	R-0/W1S	0h	Break Error Interrupt Clear Writing a 1 to this bit clears the BERIS bit in the UARTRIS register and the BEMIS bit in the UARTMIS register. Reset type: PER.RESET
8	PEIC	R-0/W1S	0h	Parity Error Interrupt Clear Writing a 1 to this bit clears the PERIS bit in the UARTRIS register and the PEMIS bit in the UARTMIS register. Reset type: PER.RESET

Table 39-17. UARTICR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	FEIC	R-0/W1S	0h	Framing Error Interrupt Clear Writing a 1 to this bit clears the FERIS bit in the UARTRIS register and the FEMIS bit in the UARTMIS register. Reset type: PER.RESET
6	RTIC	R-0/W1S	0h	Receive Time-Out Interrupt Clear Writing a 1 to this bit clears the RTRIS bit in the UARTRIS register and the RTMIS bit in the UARTMIS register. Reset type: PER.RESET
5	TXIC	R-0/W1S	0h	Transmit Interrupt Clear Writing a 1 to this bit clears the TXRIS bit in the UARTRIS register and the TXMIS bit in the UARTMIS register. Reset type: PER.RESET
4	RXIC	R-0/W1S	0h	Receive Interrupt Clear Writing a 1 to this bit clears the RXRIS bit in the UARTRIS register and the RXMIS bit in the UARTMIS register. Reset type: PER.RESET
3	RESERVED	R-0/W1S	0h	Reserved
2	RESERVED	R-0/W1S	0h	Reserved
1	RESERVED	R-0/W1S	0h	Reserved
0	RESERVED	R-0/W1S	0h	Reserved

39.5.2.14 UARTDMACTL Register (Offset = 48h) [Reset = 00000000h]

UARTDMACTL is shown in Figure 39-17 and described in Table 39-18.

Return to the Summary Table.

UART DMA Control

Figure 39-17. UARTDMACTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					DMAERR	TXDMAE	RXDMAE
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 39-18. UARTDMACTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	DMAERR	R/W	0h	DMA on Error 0 DMA receive requests are unaffected when a receive error occurs. 1 DMA receive requests are automatically disabled when a receive error occurs. Reset type: PER.RESET
1	TXDMAE	R/W	0h	Transmit DMA Enable 0 DMA for the transmit FIFO is disabled. 1 DMA for the transmit FIFO is enabled. Reset type: PER.RESET
0	RXDMAE	R/W	0h	Receive DMA Enable 0 DMA for the receive FIFO is disabled. 1 DMA for the receive FIFO is enabled. Reset type: PER.RESET

39.5.2.15 UART_GLB_INT_EN Register (Offset = 80h) [Reset = 00000000h]

UART_GLB_INT_EN is shown in Figure 39-18 and described in Table 39-19.

Return to the Summary Table.

The UART_GLB_INT_EN register is used to enable interrupt from UART to PIE

Figure 39-18. UART_GLB_INT_EN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							RESERVED
R-0h							R/W-0h

Table 39-19. UART_GLB_INT_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	RESERVED	R/W	0h	Reserved

39.5.2.16 UART_GLB_INT_FLG Register (Offset = 84h) [Reset = 00000000h]

UART_GLB_INT_FLG is shown in Figure 39-19 and described in Table 39-20.

Return to the Summary Table.

The UART_GLB_INT_FLG register contains the current status of the UART interrupt

Figure 39-19. UART_GLB_INT_FLG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							INT_FLG
R-0h							R-0h

Table 39-20. UART_GLB_INT_FLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	INT_FLG	R	0h	Global Interrupt Flag for UART INT. This bit determines whether the SINTREQUEST is generated by UART This bit can be cleared by writing a 1 to the corresponding bit in the UART_GLB_INT_CLR register. Reset type: SYSRSn

39.5.2.17 UART_GLB_INT_CLR Register (Offset = 88h) [Reset = 00000000h]

UART_GLB_INT_CLR is shown in Figure 39-20 and described in Table 39-21.

Return to the Summary Table.

The UART_GLB_INT_CLR register is used to clear the interrupt flags in UART_GLB_INT_FLG register.

Figure 39-20. UART_GLB_INT_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							INT_FLG_CLR
R-0h							R/W1C-0h

Table 39-21. UART_GLB_INT_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	INT_FLG_CLR	R/W1C	0h	Global Interrupt flag clear for UART INT. This bit is used to clear the corresponding bit in the UART_GLB_INT_FLG register. Write 1 to clear the INT_FLG bit. Writing 0 has no effect. Reset type: SYSRSn

39.5.2.18 UART9BITADDR Register (Offset = A4h) [Reset = 00000000h]

UART9BITADDR is shown in Figure 39-21 and described in Table 39-22.

Return to the Summary Table.

The UART9BITADDR register is used to write the specific address that should be matched with the receiving byte when the 9-bit Address Mask (UART9BITAMASK) is set to 0xFF. This register is used in conjunction with UART9BITAMASK to form a match for address-byte received.

Figure 39-21. UART9BITADDR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
9BITEN	RESERVED						
R/W-0h	R-0h						
7	6	5	4	3	2	1	0
ADDR							
R/W-0h							

Table 39-22. UART9BITADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	9BITEN	R/W	0h	Enable 9-Bit Mode 0 9-bit mode is disabled. 1 9-bit mode is enabled. Reset type: PER.RESET
14-8	RESERVED	R	0h	Reserved
7-0	ADDR	R/W	0h	Self Address for 9-Bit Mode This field contains the address that should be matched when UART9BITAMASK is 0xFF. Reset type: PER.RESET

39.5.2.19 UART9BITAMASK Register (Offset = A8h) [Reset = 000000FFh]

UART9BITAMASK is shown in Figure 39-22 and described in Table 39-23.

Return to the Summary Table.

The UART9BITAMASK register is used to enable the address mask for 9-bit mode. The address bits are masked to create a set of addresses to be matched with the received address byte.

Figure 39-22. UART9BITAMASK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MASK							
R-0h																								R/W-FFh							

Table 39-23. UART9BITAMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	MASK	R/W	FFh	Self Address Mask for 9-Bit Mode This field contains the address mask that creates a set of addresses that should be matched. Reset type: PER.RESET

39.5.2.20 UARTPP Register (Offset = FC0h) [Reset = 00000002h]

UARTPP is shown in Figure 39-23 and described in Table 39-24.

Return to the Summary Table.

The UARTPP register provides information regarding the properties of the UART module.

Figure 39-23. UARTPP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												MSE	MS	NB	SC
R-0h												R-0h	R-0h	R-1h	R-0h

Table 39-24. UARTPP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3	MSE	R	0h	Modem Support Extended 0 The UART module does not provide extended support for modem control. 1 The UART module provides extended support for modem control including UARTnDTR, UARTnDSR, UARTnDCD, and UARTnRI. Reset type: PER.RESET
2	MS	R	0h	Modem Support 0 The UART module does not provide support for modem control. 1 The UART module provides support for modem control including UARTnRTS and UARTnCTS. Reset type: PER.RESET
1	NB	R	1h	9-Bit Support 0 The UART module does not provide support for the transmission of 9-bit data for RS-485 support. 1 The UART module provides support for the transmission of 9-bit data for RS-485 support. Reset type: PER.RESET
0	SC	R	0h	Smart Card Support 0 The UART module does not provide smart card support. 1 The UART module provides smart card support. Reset type: PER.RESET

39.5.2.21 UARTPeriphID4 Register (Offset = FD0h) [Reset = 00000060h]

UARTPeriphID4 is shown in Figure 39-24 and described in Table 39-25.

Return to the Summary Table.

UART Peripheral Identification 4

Figure 39-24. UARTPeriphID4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PID4							
R-0h																								R-60h							

Table 39-25. UARTPeriphID4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	PID4	R	60h	UART Peripheral ID Register [7:0] Can be used by software to identify the presence of this peripheral. Reset type: PER.RESET

39.5.2.22 UARTPeriphID5 Register (Offset = FD4h) [Reset = 00000000h]

UARTPeriphID5 is shown in Figure 39-25 and described in Table 39-26.

Return to the Summary Table.

UART Peripheral Identification 5

Figure 39-25. UARTPeriphID5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PID5							
R-0h																								R-0h							

Table 39-26. UARTPeriphID5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	PID5	R	0h	UART Peripheral ID Register [15:8] Can be used by software to identify the presence of this peripheral. Reset type: PER.RESET

39.5.2.23 UARTPeriphID6 Register (Offset = FD8h) [Reset = 00000000h]

UARTPeriphID6 is shown in Figure 39-26 and described in Table 39-27.

Return to the Summary Table.

UART Peripheral Identification 6

Figure 39-26. UARTPeriphID6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PID6							
R-0h																								R-0h							

Table 39-27. UARTPeriphID6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	PID6	R	0h	UART Peripheral ID Register [23:16] Can be used by software to identify the presence of this peripheral. Reset type: PER.RESET

39.5.2.24 UARTPeriphID7 Register (Offset = FDCh) [Reset = 00000000h]

UARTPeriphID7 is shown in Figure 39-27 and described in Table 39-28.

Return to the Summary Table.

UART Peripheral Identification 7

Figure 39-27. UARTPeriphID7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PID7							
R-0h																								R-0h							

Table 39-28. UARTPeriphID7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	PID7	R	0h	UART Peripheral ID Register [31:24] Can be used by software to identify the presence of this peripheral. Reset type: PER.RESET

39.5.2.25 UARTPeriphID0 Register (Offset = FE0h) [Reset = 00000011h]

UARTPeriphID0 is shown in Figure 39-28 and described in Table 39-29.

Return to the Summary Table.

UART Peripheral Identification 0

Figure 39-28. UARTPeriphID0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PID0							
R-0h																								R-11h							

Table 39-29. UARTPeriphID0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	PID0	R	11h	UART Peripheral ID Register [7:0] Can be used by software to identify the presence of this peripheral. Reset type: PER.RESET

39.5.2.26 UARTPeriphID1 Register (Offset = FE4h) [Reset = 00000000h]

UARTPeriphID1 is shown in Figure 39-29 and described in Table 39-30.

Return to the Summary Table.

UART Peripheral Identification 1

Figure 39-29. UARTPeriphID1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PID1							
R-0h																								R-0h							

Table 39-30. UARTPeriphID1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	PID1	R	0h	UART Peripheral ID Register [15:8] Can be used by software to identify the presence of this peripheral. Reset type: PER.RESET

39.5.2.27 UARTPeriphID2 Register (Offset = FE8h) [Reset = 00000018h]

UARTPeriphID2 is shown in Figure 39-30 and described in Table 39-31.

Return to the Summary Table.

UART Peripheral Identification 2

Figure 39-30. UARTPeriphID2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PID2							
R-0h																								R-18h							

Table 39-31. UARTPeriphID2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	PID2	R	18h	UART Peripheral ID Register [23:16] Can be used by software to identify the presence of this peripheral. Reset type: PER.RESET

39.5.2.28 UARTPeriphID3 Register (Offset = FECh) [Reset = 00000001h]

UARTPeriphID3 is shown in Figure 39-31 and described in Table 39-32.

Return to the Summary Table.

UART Peripheral Identification 3

Figure 39-31. UARTPeriphID3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PID3							
R-0h																								R-1h							

Table 39-32. UARTPeriphID3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	PID3	R	1h	UART Peripheral ID Register [31:24] Can be used by software to identify the presence of this peripheral. Reset type: PER.RESET

39.5.2.29 UARTPCellID0 Register (Offset = FF0h) [Reset = 0000000Dh]

UARTPCellID0 is shown in Figure 39-32 and described in Table 39-33.

Return to the Summary Table.

UART PrimeCell Identification 0

Figure 39-32. UARTPCellID0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								CID0							
R-0h																								R-Dh							

Table 39-33. UARTPCellID0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	CID0	R	Dh	UART PrimeCell ID Register [7:0] Provides software a standard cross-peripheral identification system. Reset type: PER.RESET

39.5.2.30 UARTPCellID1 Register (Offset = FF4h) [Reset = 000000F0h]

UARTPCellID1 is shown in Figure 39-33 and described in Table 39-34.

Return to the Summary Table.

UART PrimeCell Identification 1

Figure 39-33. UARTPCellID1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								CID1							
R-0h																								R-F0h							

Table 39-34. UARTPCellID1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	CID1	R	F0h	UART PrimeCell ID Register [15:8] Provides software a standard cross-peripheral identification system. Reset type: PER.RESET

39.5.2.31 UARTPCellID2 Register (Offset = FF8h) [Reset = 00000005h]

UARTPCellID2 is shown in Figure 39-34 and described in Table 39-35.

Return to the Summary Table.

UART PrimeCell Identification 2

Figure 39-34. UARTPCellID2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								CID2							
R-0h																								R-5h							

Table 39-35. UARTPCellID2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	CID2	R	5h	UART PrimeCell ID Register [23:16] Provides software a standard cross-peripheral identification system. Reset type: PER.RESET

39.5.2.32 UARTPCellID3 Register (Offset = FFCh) [Reset = 000000B1h]

UARTPCellID3 is shown in Figure 39-35 and described in Table 39-36.

Return to the Summary Table.

UART PrimeCell Identification 3

Figure 39-35. UARTPCellID3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								CID3							
R-0h																								R-B1h							

Table 39-36. UARTPCellID3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	CID3	R	B1h	UART PrimeCell ID Register [31:24] Provides software a standard cross-peripheral identification system. Reset type: PER.RESET

39.5.3 UART_REGS_WRITE Registers

Table 39-37 lists the memory-mapped registers for the UART_REGS_WRITE registers. All register offset addresses not listed in Table 39-37 should be considered as reserved locations and the register contents should not be modified.

Table 39-37. UART_REGS_WRITE Registers

Offset	Acronym	Register Name	Protection
4h	UARTECR	UART Error Clear	

Complex bit access types are encoded to fit into small table cells. Table 39-38 shows the codes that are used for access types in this section.

Table 39-38. UART_REGS_WRITE Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

39.5.3.1 UARTECR Register (Offset = 4h) [Reset = 00000000h]

UARTECR is shown in Figure 39-36 and described in Table 39-39.

Return to the Summary Table.

The UARTECR register is the error clear register.
In addition to the UARTDR register, receive status can also be read from the UARTRSR register. If the status is read from this register, then the status information corresponds to the entry read from UARTDR prior to reading UARTRSR. The status information for overrun is set immediately when an overrun condition occurs.
The UARTRSR register cannot be written.
A write of any value to the UARTECR register clears the framing, parity, break, and overrun errors.
All the bits are cleared on reset.

Figure 39-36. UARTECR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DATA							
R-0/W-0h																								R-0/W-0h							

Table 39-39. UARTECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0/W	0h	Reserved
7-0	DATA	R-0/W	0h	Error Clear A write to this register of any data clears the framing, parity, break, and overrun flags. Reset type: PER.RESET

Chapter 40

Local Interconnect Network (LIN)

This chapter describes the local interconnect network (LIN) module. This module can also operate as a serial communications interface (SCI) port when configured in SCI/UART mode. However, since the LIN module uses a different register/bit structure, code written for this module cannot be directly ported to the standalone SCI module or vice versa.

40.1 LIN Overview

The LIN module is compliant to the LIN2.1 standard in the *LIN Specification Package*. This standard is based on the UART serial protocol. The protocol involves a single commander and one or moreresponder nodes with a message identification for multicast transmission between any network nodes.

Section 40.2 explains how the LIN module operates in SCI mode, while Section 40.3 explains how the LIN module operates in LIN mode. The register descriptions specify which register fields are applicable for each mode.

40.1.1 LIN Mode Features

When operating in LIN mode, the LIN module includes the following features:

- Compatibility with LIN 1.3 , 2.0, and 2.1 protocols
- Configurable baud rate up to 20kbps
- Two external pins: LINRX and LINTX.
- Multibuffered receive and transmit units
- Identification masks for message filtering
- Automatic commander header generation
  - Programmable synchronization break field
  - Synchronization field
  - Identifier field
- Responder Automatic Synchronization
  - Synchronization break detection
  - Optional baud rate update
  - Synchronization validation
- 2^{31} programmable transmission rates with 7 fractional bits
- Wakeup on LINRX active level from transceiver
- Automatic wake-up support
  - LINTX wake-up signal generation
  - Wake-up signal timeout
- Automatic idle bus detection
- Error detection
  - Bit error
  - Bus error
  - No-response error
  - Checksum error
  - Synchronization field error
  - Parity error
- Capability to use real-Time Direct Memory Access (RTDMA) to transmit and receive data.
- 2 interrupt lines (INT0 and INT1) with user-configurable interrupt sources:
  - Receive
  - Transmit
  - ID, error, and status
- Support for LIN 2.0 checksum
- Enhanced synchronizer finite state machine (FSM) support for frame processing
- Enhanced handling of extended frames
- Enhanced baud rate generator

40.1.2 SCI Mode Features

When operating in SCI mode, the LIN module includes the following features:

- Standard universal asynchronous receiver-transmitter (UART) communication
- Supports full-duplex or half-duplex operation
- Standard non-return-to-zero (NRZ) format
- Double-buffered receive and transmit functions
- Supports two individually enabled interrupt lines: level 0 and level 1
- Configurable frame format of three to thirteen bits per character based on the following:
  - One start bit
  - Data word length programmable from one to eight bits
  - Additional address bit in address-bit mode
  - Parity programmable for zero or one parity bit, odd or even parity
  - Stop programmable for one or two stop bits
- Asynchronous communication mode
- Two multiprocessor communication formats allow communication between more than two devices
- Sleep and wake-up functions for multiprocessor communication
- Programmable divider to support up to 2^{24} different baud rates
- Capability to use Real-Time Direct Memory Access (RTDMA) to transmit and receive data
- Error and status flags to provide detailed information about SCI events
- Multibuffer mode for receive and transmit

Note

In SCI mode, the LIN module is functionally compatible with the C2000™ SCI module, but not software compatible due to different register definitions.

The LIN module does not support UART hardware flow control. This feature can be implemented in software using a general-purpose I/O pin.

The LIN module does not support isosynchronous mode as there is no SCICLK pin.

40.1.3 Block Diagram

The LIN module is based on the SCI module with added sub-blocks to support LIN protocol.

The three major components of the SCI block are:

- **Transmitter** (TX) contains two major registers to perform the double-buffering:
  - The transmitter data buffer register (SCITD) contains data loaded by the CPU to be transferred to the shift register for transmission.
  - The transmitter shift register (SCITXSHF) loads data from the data buffer (SCITD) and shifts data onto the LINTX pin, one bit at a time.
- **Baud Clock Generator**
  - A programmable baud generator produces a baud clock scaled from the input clock VCLK
  - LIN VCLK is based on the SYSCLK frequency. VCLK input from SYSCLK and can be divided by 1, 2, or 4 using the CLK_CFG_REGS PERCLKDIVSEL.LINxCLKDIV field for each LIN module individually. By default, VCLK input is SYSCLK divided by 2
- **Receiver** (RX) contains two major registers to perform the double-buffering:
  - The receiver shift register (SCIRXSHF) shifts data in from the LINRX pin one bit at a time and transfers completed data into the receive data buffer.
  - The receiver data buffer register (SCIRD) contains received data transferred from the receiver shift register

There are separate enable and interrupt bits for the LIN receiver and transmitter. The receiver and transmitter can each be operated independently (half-duplex) or simultaneously (full duplex).

To maintain data integrity, the LIN checks the received data for breaks, parity, overrun, and framing errors. The baud rate (bits per second) is programmable to over 16 million different rates through a 24-bit baud-select register. Figure 40-1 shows the detailed diagram of the SCI block.

The LIN module is based on the SCI module with the addition of an error detector (parity calculator, checksum calculator, and bit monitor), a mask filter, a synchronizer, and a multibuffered receiver and transmitter. The SCI interface, the RTDMA control sub-blocks and the baud generator are modified for compatibility with the LIN standard. Figure 40-2 shows the LIN block diagram.

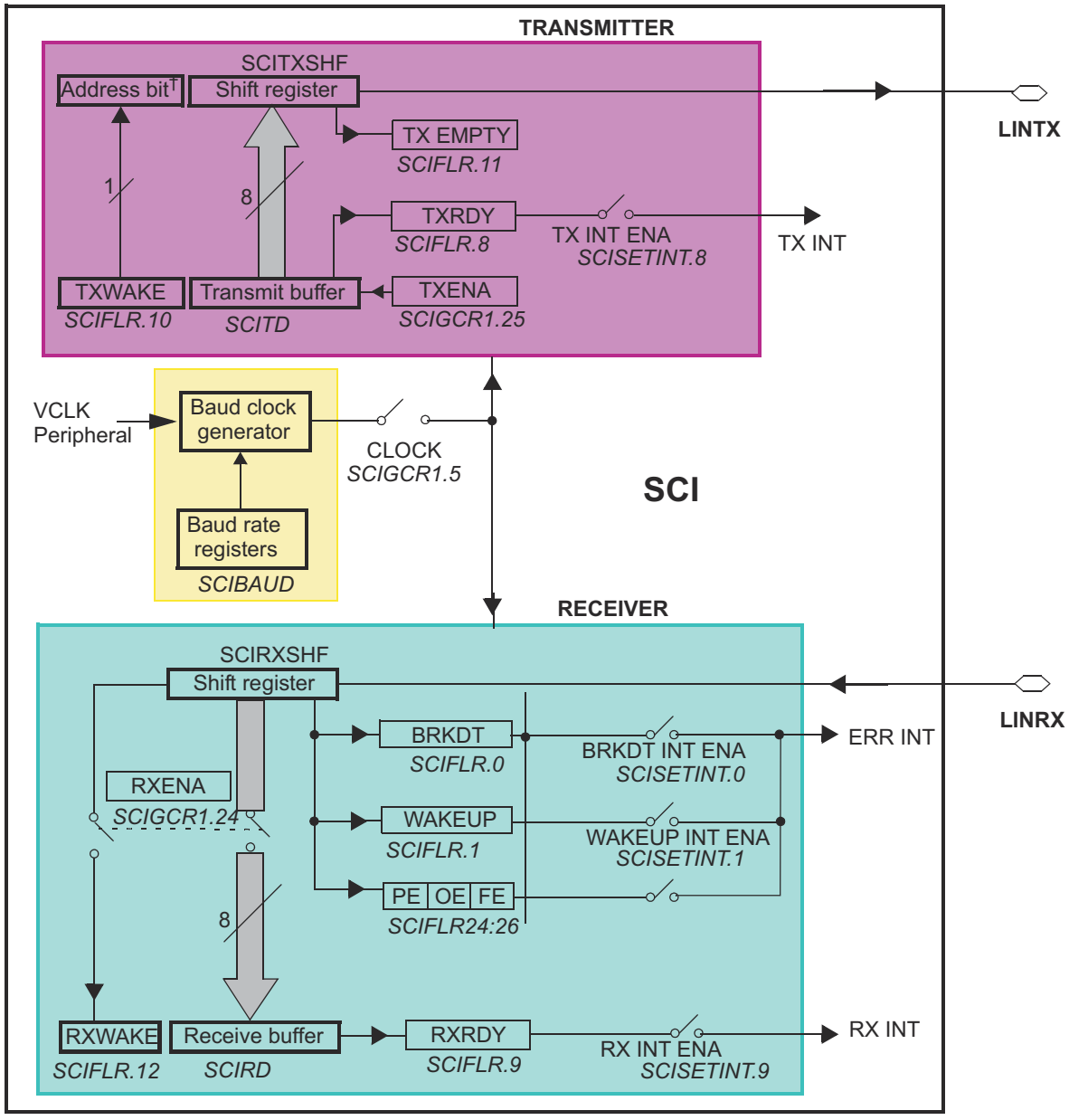


Figure 40-1. SCI Block

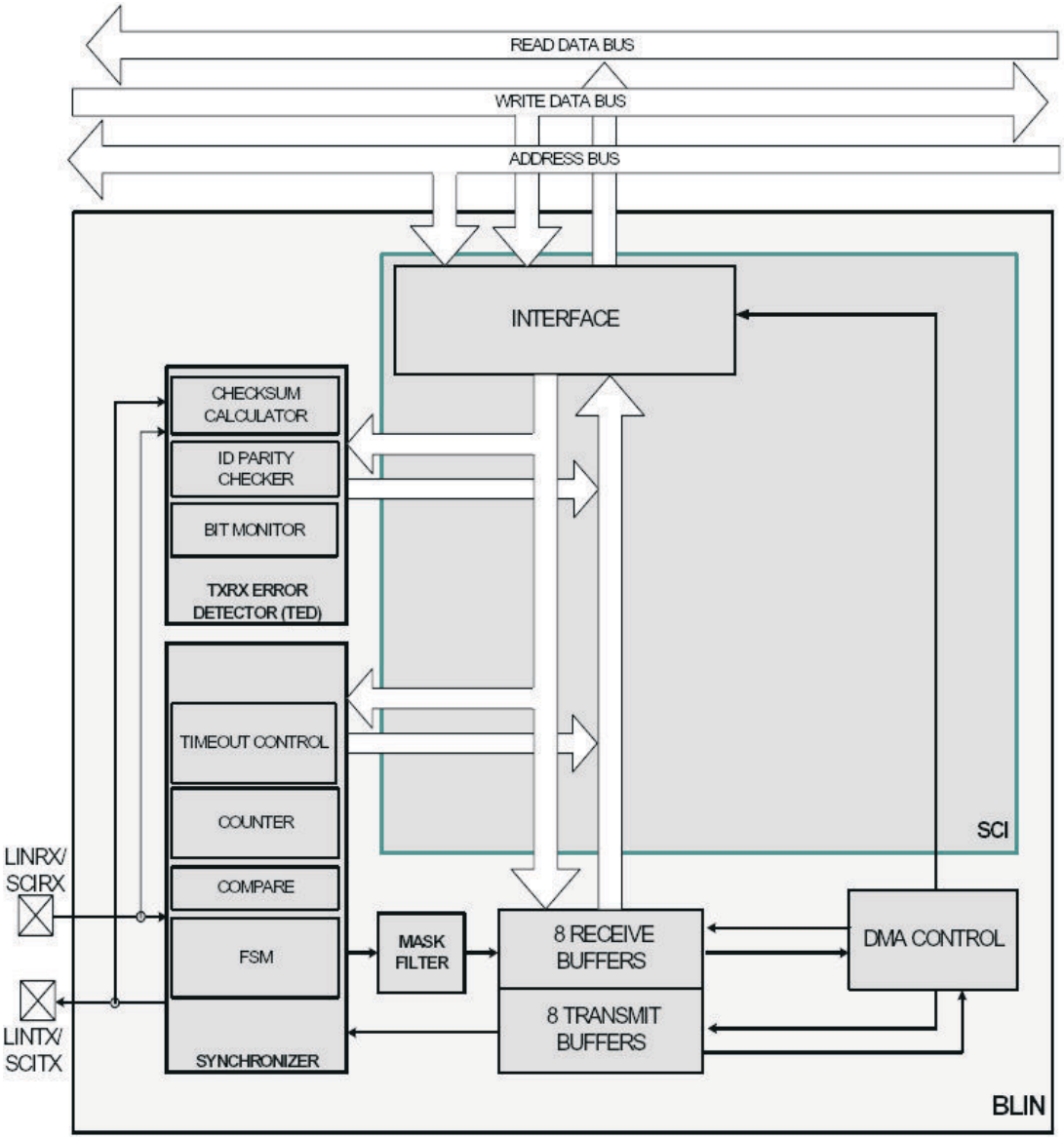


Figure 40-2. LIN Block Diagram

40.2 Serial Communications Interface Module

40.2.1 SCI Communication Formats

The SCI module can be configured to meet the requirements of many applications. Because communication formats vary depending on the specific application, many attributes of the SCI/LIN are user configurable. The configuration options are:

- SCI Frame format
- SCI Timing modes
- SCI Baud rate
- SCI Multiprocessor modes

40.2.1.1 SCI Frame Formats

The SCI uses a programmable frame format. All frames consist of the following:

- One start bit
- One to eight data bits
- Zero or one address bit
- Zero or one parity bit
- One or two stop bits

The frame format for both the transmitter and receiver is programmable through the bits in the SCIGCR1 register. Both receive and transmit data is in nonreturn to zero (NRZ) format, which means that the transmit and receive lines are at logic high when idle. Each frame transmission begins with a start bit, in which the transmitter pulls the SCI line low (logic low). Following the start bit, the frame data is sent and received least significant bit first (LSB).

An address bit is present in each frame if the SCI is configured to be in address-bit mode but is not present in any frame if the SCI is configured for idle-line mode. The format of frames with and without the address bit is illustrated in Figure 40-3.

A parity bit is present in every frame when the PARITY ENA bit is set. The value of the parity bit depends on the number of one bits in the frame and whether odd or even parity has been selected using the PARITY ENA bit. Both examples in Figure 40-3 have parity enabled.

All frames include one stop bit, which is always a high level. This high level at the end of each frame is used to indicate the end of a frame to make sure synchronization between communicating devices. Two stop bits are transmitted, if the STOP bit in SCIGCR1 register is set. The examples shown in Figure 40-3 use one stop bit per frame.

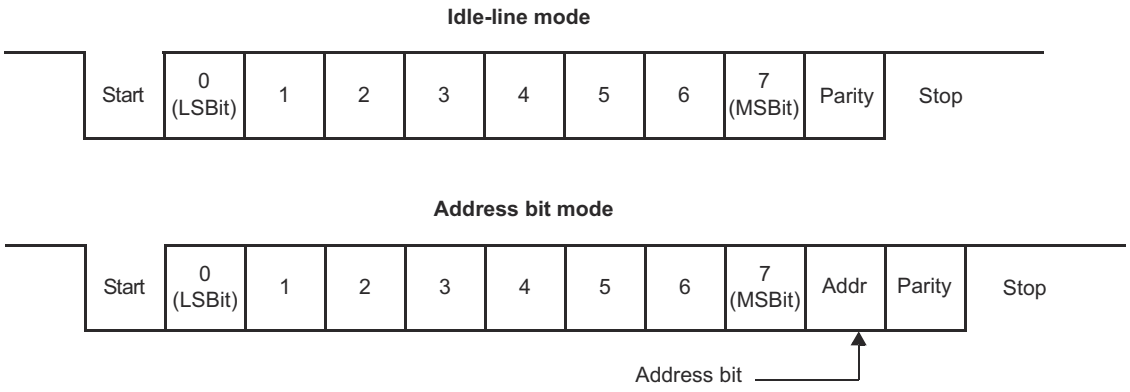


Figure 40-3. Typical SCI Data Frame Formats

40.2.1.2 SCI Asynchronous Timing Mode

The SCI can be configured to use the asynchronous timing mode using TIMING MODE bit in SCIGCR1 register.

The asynchronous timing mode uses only the receive and transmit data lines to interface with devices using the standard universal asynchronous receiver- transmitter (UART) protocol.

In the asynchronous timing mode, each bit in a frame has a duration of 16 SCI baud clock periods. Each bit therefore consists of 16 samples (one for each clock period). When the SCI is using asynchronous mode, the baud rates of all communicating devices must match as closely as possible. Receive errors result from devices communicating at different baud rates.

With the receiver in the asynchronous timing mode, the SCI detects a valid start bit if the first four samples after a falling edge on the LINRX pin are of logic level 0. As soon as a falling edge is detected on LINRX, the SCI assumes that a frame is being received and synchronizes to the bus.

To prevent interpreting noise as Start bit SCI expects LINRX line to be low for at least four contiguous SCI baud clock periods to detect a valid start bit. The bus is considered idle if this condition is not met. When a valid start bit is detected, the SCI determines the value of each bit by sampling the LINRX line value during the seventh, eighth, and ninth SCI baud clock periods. A majority vote of these three samples is used to determine the value stored in the SCI receiver shift register. By sampling in the middle of the bit, the SCI reduces errors caused by propagation delays and rise and fall times and data line noises. Figure 40-4 illustrates how the receiver samples a start bit and a data bit in asynchronous timing mode.

The transmitter transmits each bit for a duration of 16 SCI baud clock periods. During the first clock period for a bit, the transmitter shifts the value of that bit onto the LINTX pin. The transmitter then holds the current bit value on LINTX for 16 SCI baud clock periods.

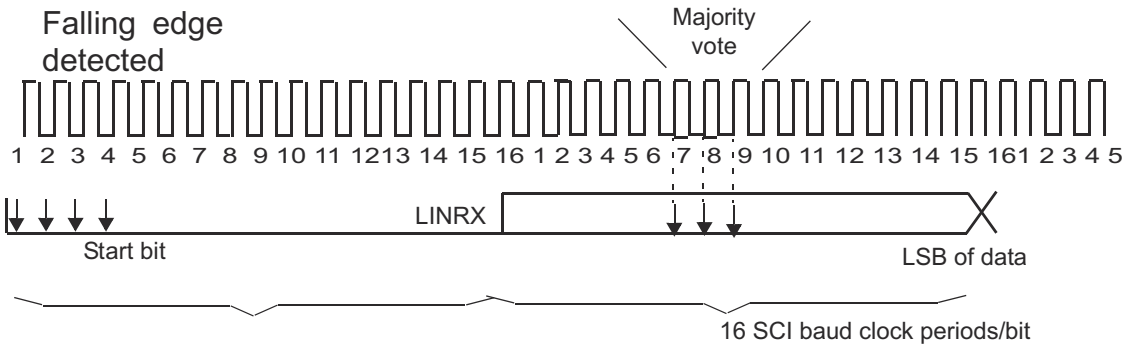


Figure 40-4. Asynchronous Communication Bit Timing

40.2.1.3 SCI Baud Rate

The SCI/LIN has an internally generated serial clock determined by the peripheral VCLK and the prescalers P and M in this register. The SCI uses the 24-bit integer prescaler P value in the BRS register to select the required baud rates. The additional 4-bit fractional divider M refines the baud rate selection.

In asynchronous timing mode, the SCI generates a baud clock according to the following formula:

$$SCICLK\ Frequency = \frac{VCLK\ Frequency}{P + 1 + \frac{M}{16}}$$

$$Asynchronous\ baud\ value = \frac{SCICLK\ Frequency}{16}$$

For P = 0,

$$Asynchronous\ baud\ value = \frac{VCLK\ Frequency}{32}$$

40.2.1.3.1 Superfractional Divider, SCI Asynchronous Mode

The superfractional divider is available in SCI asynchronous mode (idle-line and address-bit mode). Building on the 4-bit fractional divider M (BRS[27:24]), the superfractional divider uses an additional 3-bit modulating value (see Table 40-2). The bits with a 1 in the table have an additional VCLK period added to the T_{bit}. If the character length is more than 10, then the modulation table is a rolled-over version of the original table (Table 40-1), as shown in Table 40-2.

The baud rate varies over a data field to average according to the BRS[30:28] value by a "**d**" fraction of the peripheral internal clock: **0<d<1.** See Figure 40-5 for a simple Average "d' calculation based on "U" value (BRS[30:28]).

The instantaneous bit time is expressed in terms of T_{VCLK} as follows:

For all P other than 0, and all M and d (0 or 1),

$$T^{i}bit = \left[16\left(P + 1 + \frac{M}{16}\right) + d\right]T_{VCLK}$$

For P = 0, $T_{bit} = 32T_{VCLK}$

The averaged bit time is expressed in terms of T_{VCLK} as follows:

For all P other than 0, and all M and d (0<d<1),

$$T^{a}bit = \left[16\left(P + 1 + \frac{M}{16}\right) + d\right]T_{VCLK}$$

For P = 0, $T_{bit} = 32T_{VCLK}$

Table 40-1. Superfractional Bit Modulation for SCI Mode (Normal Configuration)

Normal Configuration = Start Bit + 8 Data Bits + Stop Bit										
BRS[30:28]	**Start Bit**	**D[0]**	**D[1]**	**D[2]**	**D[3]**	**D[4]**	**D[5]**	**D[6]**	**D[7]**	**Stop Bit**
0h	0	0	0	0	0	0	0	0	0	0
1h	1	0	0	0	0	0	0	0	1	0
2h	1	0	0	0	1	0	0	0	1	0
3h	1	0	1	0	1	0	0	0	1	0
4h	1	0	1	0	1	0	1	0	1	0
5h	1	1	1	0	1	0	1	0	1	1
6h	1	1	1	0	1	1	1	0	1	1
7h	1	1	1	1	1	1	1	0	1	1

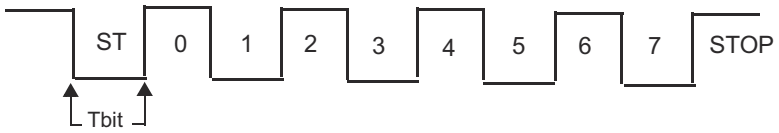
Table 40-2. Superfractional Bit Modulation for SCI Mode (Maximum Configuration)

Maximum Configuration = Start Bit + 8 Data Bits + Addr Bit + Parity Bit + Stop Bit 0 + Stop Bit 1													
BRS[30:28]	Start Bit	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]	Addr	Parity	Stop0	Stop1
0h	0	0	0	0	0	0	0	0	0	0	0	0	0
1h	1	0	0	0	0	0	0	0	1	0	0	0	0
2h	1	0	0	0	1	0	0	0	1	0	0	0	1
3h	1	0	1	0	1	0	0	0	1	0	1	0	1
4h	1	0	1	0	1	0	1	0	1	0	1	0	1
5h	1	1	1	0	1	0	1	0	1	1	1	0	1
6h	1	1	1	0	1	1	1	0	1	1	1	0	1
7h	1	1	1	1	1	1	1	0	1	1	1	1	1

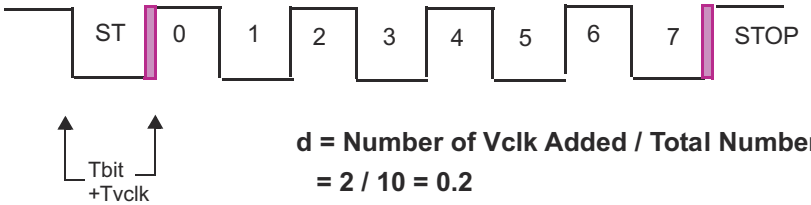
Table 40-3. SCI Mode (Minimum Configuration)

Minimum Configuration = Start Bit + 1 Data Bit + Stop Bit			
BRS[30:28]	Start Bit	D[0]	Stop Bit
0h	0	0	0
1h	1	0	0
2h	1	0	0
3h	1	0	1
4h	1	0	1
5h	1	1	1
6h	1	1	1
7h	1	1	1

Normal Data Frame with BRS[31:28] = 0



Normal Data Frame with BRS[31:28] = 1



d = Number of Vclk Added / Total Number of Bits
= 2 / 10 = 0.2

Figure 40-5. Superfractional Divider Example

40.2.1.4 SCI Multiprocessor Communication Modes

In some applications, the SCI can be connected to more than one serial communication device. In such a multiprocessor configuration, several frames of data can be sent to all connected devices or to an individual device. In the case of data sent to an individual device, the receiving devices must determine when the devices are being addressed. When a message is not intended for them, the devices can ignore the following data. When only two devices make up the SCI network, addressing is not needed, so multiprocessor communication schemes are not required.

SCI supports two multiprocessor communication modes which can be selected using COMM MODE bit:

- Idle-Line Mode
- Address Bit Mode

When the SCI is not used in a multiprocessor environment, software can consider all frames as data frames. In this case, the only distinction between the idle-line and address-bit modes is the presence of an extra bit (the address bit) in each frame sent with the address-bit protocol.

The SCI allows full-duplex communication where data can be sent and received using the transmit and receive pins simultaneously. However, the protocol used by the SCI assumes that only one device transmits data on the same bus line at any one time. No arbitration is done by the SCI.

40.2.1.4.1 Idle-Line Multiprocessor Modes

In idle-line multiprocessor mode, a frame that is preceded by an idle period (10 or more idle bits) is an address frame. A frame that is preceded by fewer than 10 idle bits is a data frame. Figure 40-6 illustrates the format of several blocks and frames with idle-line mode.

There are two ways to transmit an address frame using idle-line mode:

Method 1: In software, deliberately leave an idle period between the transmission of the last data frame of the previous block and the address frame of the new block.

Method 2: Configure the SCI to automatically send an idle period between the last data frame of the previous block and the address frame of the new block.

Although Method 1 is only accomplished by a delay loop in software, Method 2 can be implemented by using the transmit buffer and the TXWAKE bit in the following manner:

Step 1: Write a 1 to the TXWAKE bit.

Step 2: Write a dummy data value to the SCITD register. This triggers the SCI to begin the idle period as soon as the transmitter shift register is empty.

Step 3: Wait for the SCI to clear the TXWAKE flag.

Step 4: Write the address value to SCITD.

As indicated by Step 3, software can wait for the SCI to clear the TXWAKE bit. However, the SCI clears the TXWAKE bit at the same time the SCI sets TXRDY (that is, transfers data from SCITD into SCITXSHF). Therefore, if the TX INT ENA bit is set, the transfer of data from SCITD to SCITXSHF causes an interrupt to be generated at the same time that the SCI clears the TXWAKE bit. If this interrupt method is used, software is not required to poll the TXWAKE bit waiting for the SCI to clear the bit.

When idle-line multiprocessor communications are used, software must make sure that the idle time exceeds 10 bit periods before addresses (using one of the methods mentioned above), and software must also make sure that data frames are written to the transmitter quickly enough to be sent without a delay of 10 bit periods between frames. Failure to comply with these conditions results in data interpretation errors by other devices receiving the transmission.

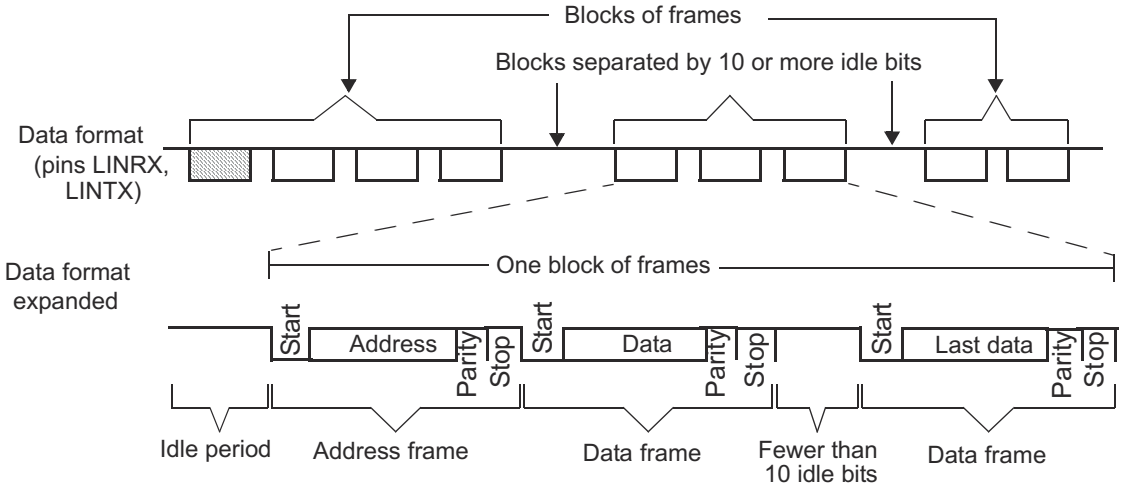


Figure 40-6. Idle-Line Multiprocessor Communication Format

40.2.1.4.2 Address-Bit Multiprocessor Mode

In the address-bit protocol, each frame has an extra bit immediately following the data field called an address bit. A frame with the address bit set to 1 is an address frame; a frame with the address bit set to 0 is a data frame. The idle period timing is irrelevant in this mode. Figure 40-7 illustrates the format of several blocks and frames with the address-bit mode.

When address-bit mode is used, the value of the TXWAKE bit is the value sent as the address bit. To send an address frame, software must set the TXWAKE bit. This bit is cleared as the contents of the SCITD are shifted from the TXWAKE register so that all frames sent are data except when the TXWAKE bit is written as a 1.

No dummy write to SCITD is required before an address frame is sent in address-bit mode. The first byte written to SCITD after the TXWAKE bit is written to 1 is transmitted with the address bit set when address-bit mode is used.

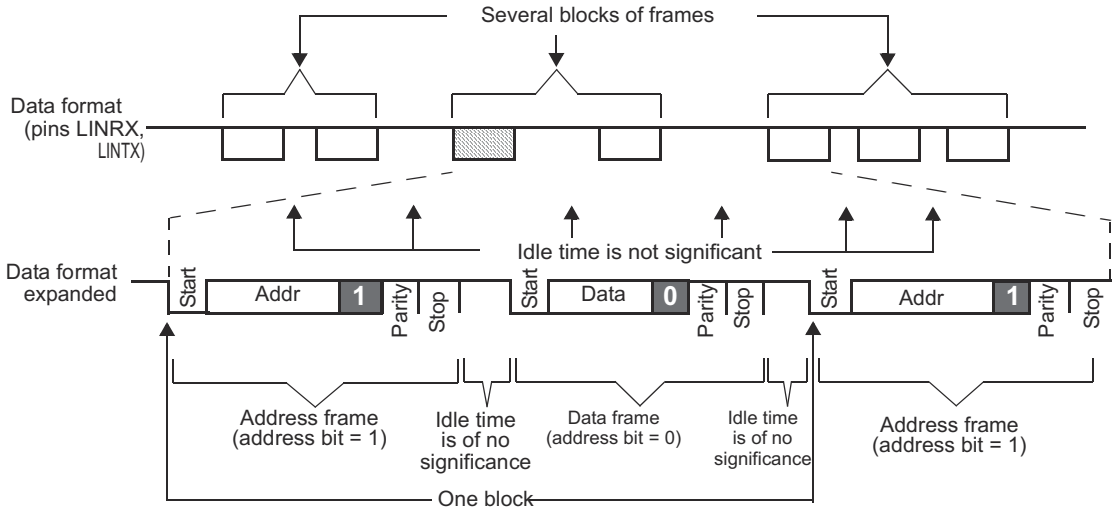


Figure 40-7. Address-Bit Multiprocessor Communication Format

40.2.1.5 SCI Multibuffered Mode

To reduce CPU load when receiving or transmitting data in interrupt mode or RTDMA mode, the SCI/LIN module has eight separate receive and transmit buffers. Multibuffered mode is enabled by setting the MBUF MODE bit.

The multibuffer 3-bit counter counts the data bytes transferred from the SCIRXSHF register to the RDy receive buffers and TDy transmit buffers register to SCITXSHF register. The 3-bit compare register contains the number of data bytes expected to be received or transmitted. the LENGTH value in SCIFORMAT register indicates the expected length and is used to load the 3-bit compare register.

A receive interrupt (RX interrupt; see the SCIINTVECT0 and SCIINTVECT1 registers), and a receive ready RXRDY flag set in SCIFLR register, as well as a RTDMA request (RXDMA) can occur after receiving a response if there are no response receive errors for the frame (such as, there is, frame error, and overrun error).

A transmit interrupt (TX interrupt), and a transmit ready flag (TXRDY flag in SCIFLR register), and a RTDMA request (TXDMA) can occur after transmitting a response.

Figure 40-8 and Figure 40-9 show the receive and transmit multibuffer functional block diagram, respectively.

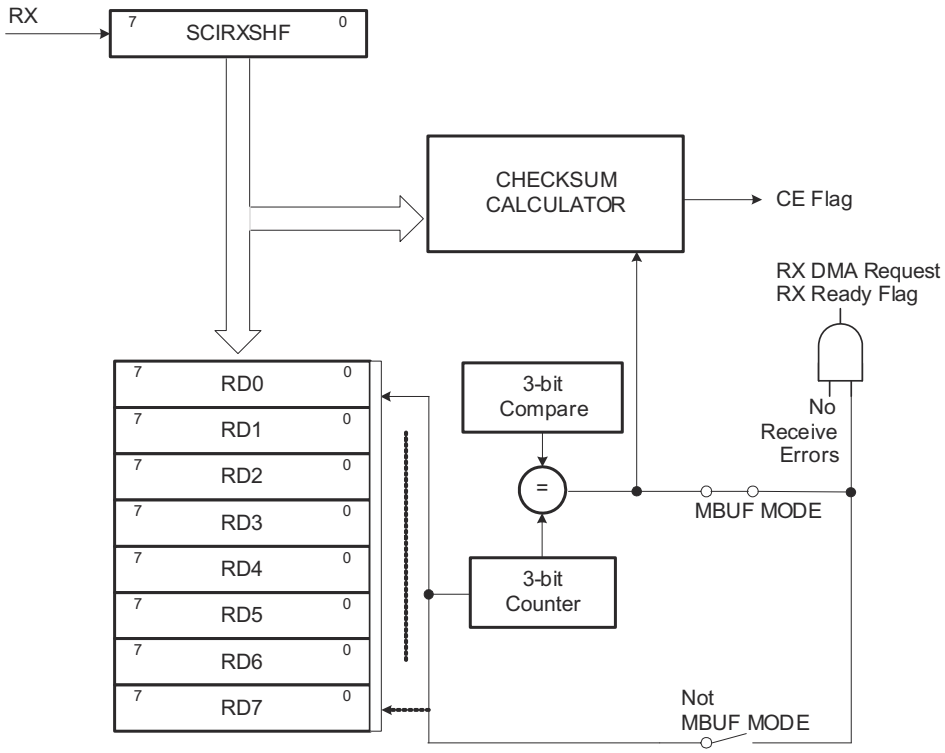


Figure 40-8. Receive Buffers

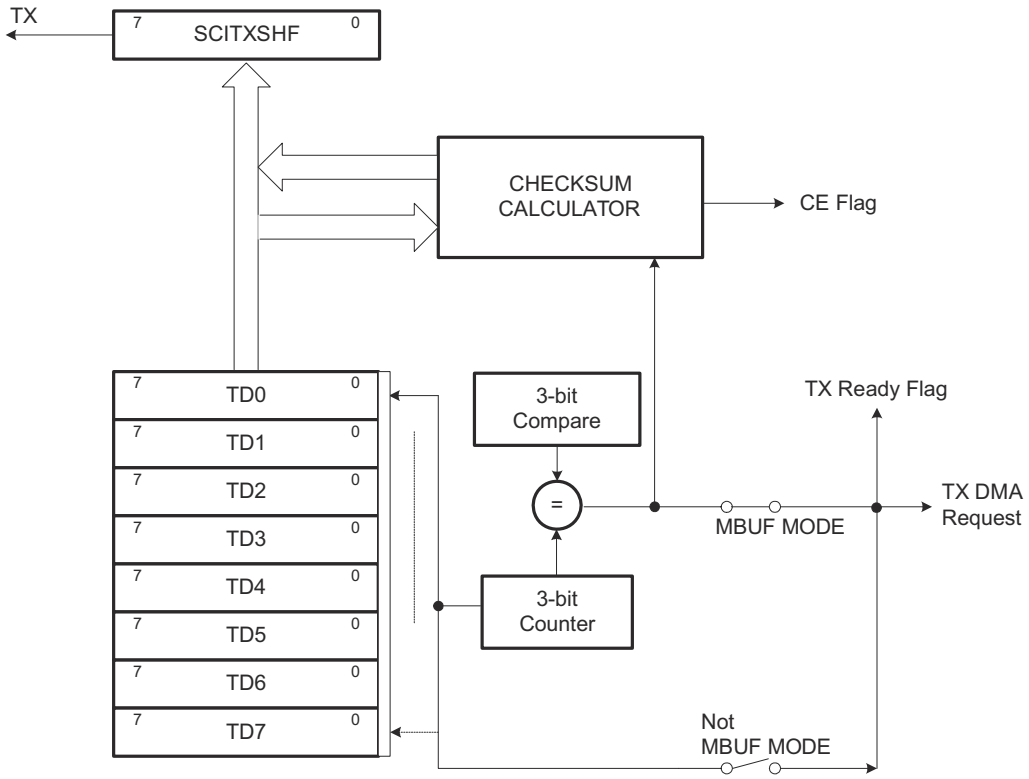


Figure 40-9. Transmit Buffers

40.2.2 SCI Interrupts

The SCI/LIN module has two interrupt lines, level 0 and level 1, to the vectored interrupt manager (VIM) module (see Figure 40-10). Two offset registers SCIINTVECT0 and SCIINTVECT1 determine which flag triggered the interrupt according to the respective priority encoders. Each interrupt condition has a bit to enable/disable the interrupt in the SCISETINT and SCICLRINT registers, respectively.

Each interrupt also has a bit that can be set as interrupt level 0(INT0) or as interrupt level 1(INT1). By default, interrupts are in interrupt level 0. SCISETINTLVL sets a given interrupt to level1. SCICLEARINTLVL resets a given interrupt level to the default level 0.

The interrupt vector registers SCIINTVECT0 and SCIINTVECT1 return the vector of the pending interrupt line INT0 or INT1. If more than one interrupt is pending, the interrupt vector register holds the highest priority interrupt.

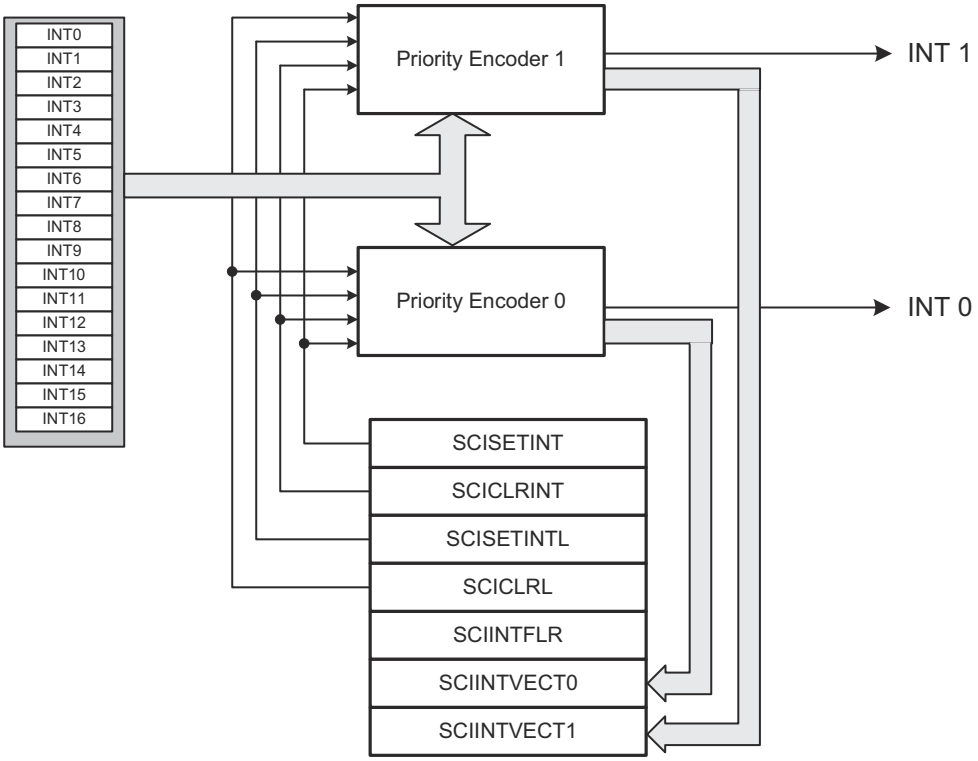


Figure 40-10. General Interrupt Scheme

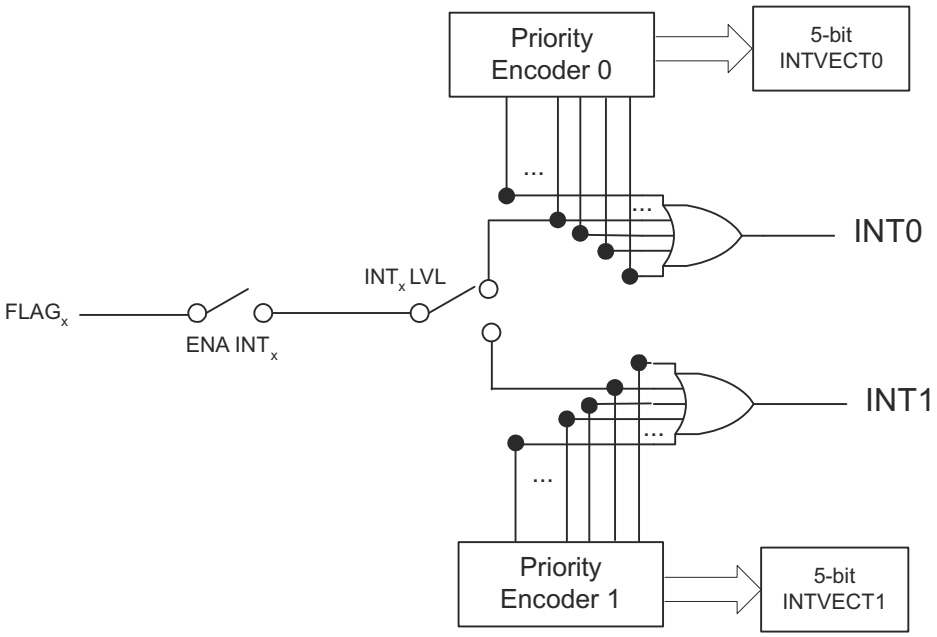


Figure 40-11. Interrupt Generation for Given Flags

40.2.2.1 Transmit Interrupt

To use transmit interrupt functionality, SETTXINT bit must be enabled and SET_TX_DMA bit must be cleared in the SCISETINT register. The transmit ready (TXRDY) flag is set when the SCI transfers the contents of SCITD/TDy to the shift register, SCITXSHF. The TXRDY flag indicates that SCITD/TDy is ready to be loaded with more data. In addition, the SCI sets the TX EMPTY bit if both the SCITD/TDy and SCITXSHF registers are empty. If the SETTXINT bit is set, then a transmit interrupt is generated when the TXRDY flag goes high. The transmit interrupt is not generated immediately after setting the SETTXINT bit unlike the transmit RTDMA request. The transmit interrupt is generated only after the first transfer from SCITD/TDy to SCITXSHF, that is first data has to be written to SCITD/TDy before any interrupt gets generated. To transmit further data, data can be written to SCITD/TDy in the transmit interrupt service routine.

Writing data to the SCITD/TDy register clears the TXRDY bit. When this data has been moved to the SCITXSHF register, the TXRDY bit is set again. The interrupt request can be suspended by setting the CLRTXINT bit in the SCICLEARINT register; however, when the SETTXINT bit is again set to 1, the TXRDY interrupt is asserted again. The transmit interrupt request can be eliminated until the next series of values is written to SCITD/TDy, by disabling the transmitter using the TXENA bit, by a software reset SWnRST, or by a device hardware reset.

40.2.2.2 Receive Interrupt

The receive ready (RXRDY) flag is set when the SCI transfers newly received data from SCIRXSHF to SCIRD/RDy. The RXRDY flag therefore indicates that the SCI has new data to be read. Receive interrupts are enabled by the SETRXINT bit in the SCISETINT register. If the SETRXINT is set when the SCI sets the RXRDY flag, then a receive interrupt is generated. The received data can be read in the Interrupt Service routine.

On a device with a RTDMA controller, the SET_RX_DMA bit in the SCISETINT register must be cleared to select interrupt functionality.

40.2.2.3 WakeUp Interrupt

SCI sets the WAKEUP flag if bus activity on the RX line either prevents power-down mode from being entered, or RX line activity causes an exit from power-down mode. If enabled (SCISETINT.SETWAKEUPINT is set), the wakeup interrupt is triggered once the WAKEUP flag in the SCIFLR register is set.

40.2.2.4 Error Interrupts

The following error detections are supported with an interrupt by the SCI module:

- Parity errors (PE)
- Frame errors (FE)
- Break Detect errors (BRKDT)
- Overrun errors (OE)
- Bit errors (BE)

There are 16 interrupt sources in the SCI/LIN module. In SCI mode, 8 interrupts are supported, as listed in Table 40-4.

If all of these errors (PE, FE, BRKDT, OE, BE) are flagged, an interrupt for the flagged errors is generated if enabled. A message is valid for both the transmitter and the receiver, if there is no error detected until the end of the frame. Each of these flags is located in the receiver status (SCIFLR) register (Table 40-5 and Table 40-6).

Table 40-4. SCI/LIN Interrupts

Offset(1)	Interrupt	Applicable to SCI	Applicable to LIN
0	No interrupt	-	-
1	Wakeup	Yes	Yes
2	Inconsistent-sync-field error (ISFE)	No	Yes
3	Parity error (PE)	Yes	Yes
4	ID	No	Yes
5	Physical bus error (PBE)	No	Yes
6	Frame error (FE)	Yes	Yes
7	Break detect (BRKDT)	Yes	No
8	Checksum error (CE)	No	Yes
9	Overrun error (OE)	Yes	Yes
10	Bit error (BE)	Yes	Yes
11	Receive	Yes	Yes
12	Transmit	Yes	Yes
13	No-response error (NRE)	No	Yes
14	Timeout after wakeup signal (150ms)	No	Yes
15	Timeout after three wakeup signals (1.5s)	No	Yes
16	Timeout (Bus Idle, 4s)	No	Yes

(1) Offset 1 is the highest priority. Offset 16 is the lowest priority.

Table 40-5. SCI Receiver Status Flags

SCI Flag	Register	Bit	Value After Reset(1)
CE	SCIFLR	29	0
ISFE	SCIFLR	28	0
NRE	SCIFLR	27	0
FE	SCIFLR	26	0
OE	SCIFLR	25	0
PE	SCIFLR	24	0
RXWAKE	SCIFLR	12	0
RXRDY	SCIFLR	9	0
BUSY	SCIFLR	3	0
IDLE	SCIFLR	2	1
WAKEUP	SCIFLR	1	0
BRKDT	SCIFLR	0	0

(1) The flags are frozen with the reset value while SWnRST = 0.

Table 40-6. SCI Transmitter Status Flags

SCI Flag	Register	Bit	Value After Reset(1)
BE	SCIFLR	31	0
PBE	SCIFLR	30	0
TXWAKE	SCIFLR	10	0
TXEMPTY	SCIFLR	11	1
TXRDY	SCIFLR	8	1

(1) The flags are frozen with the reset value while SWnRST = 0.

40.2.3 SCI RTDMA Interface

RTDMA requests for receive (RXDMA request) and transmit (TXDMA request) are available for the SCI/LIN module. The RTDMA must be configured to transfer to/from the SCITD/SCIRD register if multibuffer mode is disabled (MBUFMODE in the SCIGCR1 register is cleared), and to/from the TDy/RDy registers if multibuffer mode is enabled (MBUFMODE in the SCIGCR1 register is set.)

40.2.3.1 Receive RTDMA Requests

This RTDMA functionality is enabled/disabled by the CPU using the SETRXDMA/CLRRXDMA bits, respectively.

In multibuffered SCI mode with RTDMA enabled, the receiver loads the RDy buffers for each received character. RXDMA request is triggered once the last character of the programmed number of characters (LENGTH) are received and copied to the corresponding RDy buffer successfully.

If the multibuffer option is disabled, then RTDMA requests are generated on a byte-per-byte basis.

In multiprocessor mode, the SCI can generate receiver interrupts for address frames and RTDMA requests for data frames or RTDMA requests for both. This is controlled by the SET_RX_DMA_ALL bit.

In multiprocessor mode with the SLEEP bit set, no RTDMA request is generated for received data frames. The software must clear the SLEEP bit before data frames can be received.

40.2.3.2 Transmit RTDMA Requests

RTDMA functionality is enabled/disabled by the CPU with SET_TX_DMA/CLR_TX_DMA bits, respectively.

In multibuffered SCI mode once TXRDY bit is set or after a transmission of programmed number of characters (LENGTH) (up to eight data bytes stored in the transmit buffers (TDy) in the LINTD0 and LINTD1 registers), a RTDMA request is generated to reload the transmit buffer for the next transmission. If the multibuffer option is disabled, then RTDMA requests are generated on a byte-per-byte basis.

40.2.4 SCI Configurations

Before the SCI sends or receives data, the SCI registers can be properly configured. Upon power-up or a system-level reset, each bit in the SCI registers is set to a default state. The registers are writable only after the RESET bit in the SCIGCR0 register is set to 1. Of particular importance is the SWnRST bit in the SCIGCR1 register. The SWnRST is an active-low bit initialized to 0 and keeps the SCI in a reset state until the bit is programmed to 1. Therefore, all SCI configuration can be completed before a 1 is written to the SWnRST bit.

The following list details the configuration steps that software can perform prior to the transmission or reception of data. As long as the SWnRST bit is cleared to 0 the entire time that the SCI is being configured, the order in which the registers are programmed is not important.

- Enable SCI by setting the RESET bit to 1.
- Clear the SWnRST bit to 0 before SCI is configured.
- Select the desired frame format by programming the SCIGCR1 register.
- Set both the RX FUNC and TX FUNC bits in SCIPIO0 to 1 to configure the LINRX and LINTX pins for SCI functionality.
- Select the baud rate to be used for communication by programming the BRS register.
- Set the CLOCK bit in SCIGCR1 to 1 to select the internal clock.
- Set the CONT bit in SCIGCR1 to 1 to make SCI not halt for an emulation breakpoint until the current reception or transmission is complete (this bit is used only in an emulation environment).
- Set the LOOP BACK bit in SCIGCR1 to 1 to connect the transmitter to the receiver internally (this feature is used to perform a self-test).
- Set the RXENA bit in SCIGCR1 to 1, if data is to be received.
- Set the TXENA bit in SCIGCR1 to 1, if data is to be transmitted.
- Set the SWnRST bit to 1 after SCI is configured.
- Perform receiving or transmitting data (see Section 40.2.4.1 or Section 40.2.4.2).

40.2.4.1 Receiving Data

The SCI receiver is enabled to receive messages, if both the RX FUNC bit and the RXENA bit are set to 1. If the RX FUNC bit is not set, the LINRX pin functions as a general-purpose I/O pin rather than as an SCI function pin.

SCI module can receive data in one of the following modes:

- Single-Buffer (Normal) Mode
- Multibuffer Mode

After a valid idle period is detected, data is automatically received as the data arrives on the LINRX pin.

40.2.4.1.1 Receiving Data in Single-Buffer Mode

Single-buffer mode is selected when the MBUFMODE bit in SCIGCR1 is cleared to 0. In this mode, SCI sets the RXRDY bit when the SCI transfers newly received data from SCIRXSHF to SCIRD. The RXRDY bit is cleared after the new data in SCIRD has been read. Also, as data is transferred from SCIRXSHF to SCIRD, the FE, OE, or PE flags are set if any of these error conditions were detected in the received data. These error conditions are supported with configurable interrupt capability. The wakeup and break-detect status bits are also set if one of these errors occurs, but the bits do not necessarily occur at the same time that new data is being loaded into SCIRD.

You can receive data by:

1. Polling the Receive Ready Flag
2. Receive Interrupt
3. RTDMA

In polling method, software can poll for the RXRDY bit and read the data from the SCIRD register once the RXRDY bit is set high. The CPU is unnecessarily overloaded by selecting the polling method. To avoid this, you can use either the interrupt or RTDMA method. To use the interrupt method, set the SETRXINT bit. To use the RTDMA method, set the SET_RX_DMA bit. Either an interrupt or a RTDMA request is generated the moment the RXRDY bit is set.

40.2.4.1.2 Receiving Data in Multibuffer Mode

Multibuffer mode is selected when the MBUFMODE bit in SCIGCR1 is set to 1. In this mode, SCI sets the RXRDY bit after receiving the programmed number of data in the receive buffer, the complete frame. The error condition detection logic is similar to the single-buffer mode, except that this logic monitors for the complete frame. Like single-buffer mode, use the polling, RTDMA, or interrupt method to read the data. The RXRDY bit is automatically cleared after the new data in SCIRD has been read.

40.2.4.2 Transmitting Data

The SCI transmitter is enabled if both the TXFUNC bit and the TXENA bit are set to 1. If the TXFUNC bit is not set, the LINTX pin functions as a general-purpose I/O pin rather than as an SCI function pin. Any value written to the SCITD/TDy before TXENA is set to 1 is not transmitted. Both of these control bits allow for the SCI transmitter to be held inactive independently of the receiver.

The SCI module can transmit data in one of the following modes:

- Single-Buffer (Normal) Mode
- Multibuffered or Buffered SCI Mode

40.2.4.2.1 Transmitting Data in Single-Buffer Mode

Single-buffer mode is selected when the MBUFMODE bit in SCIGCR1 is cleared to 0. In this mode, the SCI waits for data to be written to SCITD, transfers the data to SCITXSHF, and transmits the data. The TXRDY and TXEMPTY bits indicate the status of the transmit buffers. That is, when the transmitter is ready for data to be written to SCITD, the TXRDY bit is set. Additionally, if both SCITD and SCITXSHF are empty, then the TXEMPTY bit is also set.

You can transmit data by:

1. Polling the Transmit Ready Flag
2. Transmit Interrupt
3. RTDMA

With the polling method, software can poll for the TXRDY bit to go high before writing the data to the SCITD register. The CPU is unnecessarily overloaded by selecting the polling method. To avoid this, you can use the interrupt or RTDMA method. To use the interrupt method, the SETTXINT bit is set. To use the RTDMA method, the SET_TX_DMA bit is set. Either an interrupt or a RTDMA request is generated the moment the TXRDY bit is set. When the SCI has completed transmission of all pending frames, the SCITXSHF register and SCITD are empty, the TXRDY bit is set, and an interrupt/RTDMA request is generated, if enabled. Because all data has been transmitted, the interrupt/RTDMA request must be halted. This can either be done by disabling the transmit interrupt (CLRTXINT)/RTDMA request (CLRTXDMA bit), or by disabling the transmitter (clear TXENA bit).

Note

The TXRDY flag cannot be cleared by reading the corresponding interrupt offset in the SCIINTVECT0 or SCIINTVECT1 register.

40.2.4.2.2 Transmitting Data in Multibuffer Mode

Multibuffer mode is selected when the MBUFMODE bit in SCIGCR1 is set to 1. Like single-buffer mode, you can use the polling, RTDMA, or interrupt method to write the data to be transmitted. The transmitted data has to be written to the SCITD registers. The SCI waits for data to be written to the SCITD register and then transfers the programmed number of bytes to SCITXSHF to transmit one by one automatically.

40.2.5 SCI Low-Power Mode

The SCI/LIN can be put in either local or global low-power mode. Global low-power mode is asserted by the system and is not controlled by the SCI/LIN. During global low-power mode, all clocks to the SCI/LIN are turned off so the module is completely inactive.

Local low-power mode is asserted by setting the POWERDOWN bit; setting this bit stops the clocks to the SCI/LIN internal logic and the module registers. Setting the POWERDOWN bit causes the SCI to enter local low-power mode and clearing the POWERDOWN bit causes SCI/LIN to exit from local low-power mode. All the registers are accessible during local power-down mode as any register access enables the clock to SCI for that particular access alone.

The wakeup interrupt is used to allow the SCI to exit low-power mode automatically when a low level is detected on the LINRX pin and also this clears the POWERDOWN bit. If wakeup interrupt is disabled, then the SCI/LIN immediately enters low-power mode whenever it is requested and also any activity on the LINRX pin does not cause the SCI to exit low-power mode.

Note

Enabling Local Low-Power Mode During Receive and Transmit

If the wakeup interrupt is enabled and low-power mode is requested while the receiver is receiving data, then the SCI immediately generates a wakeup interrupt to clear the powerdown bit and prevents the SCI from entering low-power mode and thus completes the current reception. Otherwise, if the wakeup interrupt is disabled, then the SCI completes the current reception and then enters the low-power mode.

40.2.5.1 Sleep Mode for Multiprocessor Communication

When the SCI receives data and transfers that data from SCIRXSHF to SCIRD, the RXRDY bit is set and if SETRXINT is set, the SCI also generates an interrupt. The interrupt triggers the CPU to read the newly received frame before another one is received. In multiprocessor communication modes, this default behavior can be enhanced to provide selective indication of new data. When the SCI receives an address frame that does not match the address, the device can ignore the data following this non-matching address until the next address frame by using sleep mode. Sleep mode can be used with both idle-line and address-bit multiprocessor modes.

If sleep mode is enabled by the SLEEP bit, then the SCI transfers data from SCIRXSHF to SCIRD only for address frames. Therefore, in sleep mode, all data frames are assembled in the SCIRXSHF register without being shifted into the SCIRD and without initiating a receive interrupt or RTDMA request. Upon reception of an address frame, the contents of the SCIRXSHF are moved into SCIRD, and the software must read SCIRD and determine if the SCI is being addressed by comparing the received address against the address previously set in the software and stored somewhere in memory (the SCI does not have hardware available for address comparison). If the SCI is being addressed, the software must clear the SLEEP bit so that the SCI loads SCIRD with the data of the data frames that follow the address frame.

When the SCI has been addressed and sleep mode has been disabled (in software) to allow the receipt of data, the SCI can check the RXWAKE bit (SCIFLR.12) to determine when the next address has been received. The bit is set to 1, if the current value in SCIRD is an address; the bit is set to 0, if SCIRD contains data. If the RXWAKE bit is set, then software can check the address in SCIRD against the address. If SCIRD is still being addressed, then sleep mode can remain disabled; otherwise, the SLEEP bit can be set again.

Following is a sequence of events typical of sleep mode operation:

- The SCI is configured and both sleep mode and receive actions are enabled.
- An address frame is received and a receive interrupt is generated.
- Software compares the received address frame against that set by software and determines that the SCI is not being addressed, so the value of the SLEEP bit is not changed.
- Several data frames are shifted into SCIRXSHF, but no data is moved to SCIRD and no receive interrupts are generated.
- A new address frame is received and a receive interrupt is generated.
- Software compares the received address frame against that set by software and determines that the SCI is being addressed and clears the SLEEP bit.
- Data shifted into SCIRXSHF is transferred to SCIRD, and a receive interrupt is generated after each data frame is received.
- In each interrupt routine, software checks RXWAKE to determine if the current frame is an address frame.
- Another address frame is received, RXWAKE is set, software determines that the SCI is not being addressed and sets the SLEEP bit back to 1. No receive interrupts are generated for the data frames following this address frame.

By ignoring data frames that are not intended for the device, fewer interrupts are generated. Otherwise, these interrupts require CPU intervention to read data that is of no significance to this specific device. Using sleep mode can help free some CPU resources.

Except for the RXRDY flag, the SCI continues to update the receiver status flags (see Table 40-5) while sleep mode is active. In this way, if an error occurs on the receive line, an application can immediately respond to the error and take the appropriate corrective action.

Because the RXRDY bit is not updated for data frames when sleep mode is enabled, the SCI can enable sleep mode and use a polling algorithm if desired. In this case, when RXRDY is set, software knows that a new address has been received. If the SCI is not being addressed, then the software can not change the value of the SLEEP bit and can continue to poll RXRDY.

40.3 Local Interconnect Network Module

40.3.1 LIN Communication Formats

The SCI/LIN module can be used in LIN mode or SCI mode. The enhancements for baud generation, RTDMA controls, and additional receive/transmit buffers necessary for LIN mode operation are also part of the enhanced buffered SCI module. LIN mode is selected by enabling the LINMODE bit in SCIGCR1 register.

Note

The SCI/LIN is built around the SCI platform and uses a similar sampling scheme: 16 samples for each bit with majority vote on samples 8, 9, and 10. For the START bit, the first three samples are used.

The SCI/LIN control registers are located at the SCI/LIN base address. For a detailed description of each register, see Section 40.7.

40.3.1.1 LIN Standards

For compatibility with LIN2.0 standard the following additional features are implemented over LIN1.3:

1. Support for LIN 2.0 checksum
2. Enhanced synchronizer FSM support for frame processing
3. Enhanced handling of extended frames
4. Enhanced baud rate generator
5. Update wakeup/go to sleep

The LIN module covers the CPU performance-consuming features, defined in the *LIN Specification Package* Revision 1.3 and 2.0 by hardware. The Commander Mode of LIN module is compatible with LIN 2.1 standard.

40.3.1.2 Message Frame

The LIN protocol defines a message frame format, shown in Figure 40-12. Each frame includes one commander header, one response, one in-frame response space, and inter-byte spaces. In-frame-response and inter-byte spaces can be 0.

There is no arbitration in the definition of the LIN protocol; therefore, multiple responder nodes responding to a header can be detected as an error.

The LIN bus is a single-channel wired-AND bus. The bus has a binary level: either dominant for a value of 0 or recessive for a value of 1.

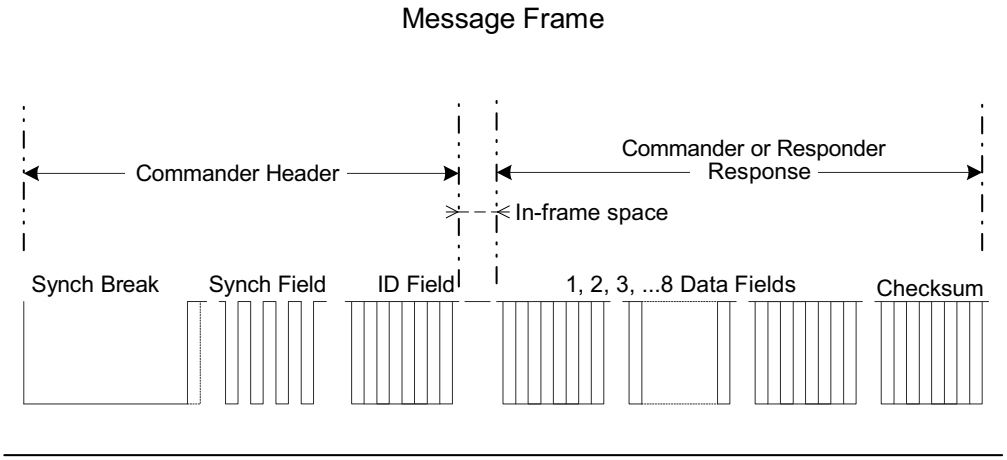


Figure 40-12. LIN Protocol Message Frame Format: Commander Header and Responder Peripheral Response

40.3.1.2.1 Message Header

The header of a message is initiated by a commander (see Figure 40-13) and consists of a three field-sequence:

- The synchronization break field signaling the beginning of a message
- The synchronization field conveying bit rate information of the LIN bus
- The identification field denoting the content of a message

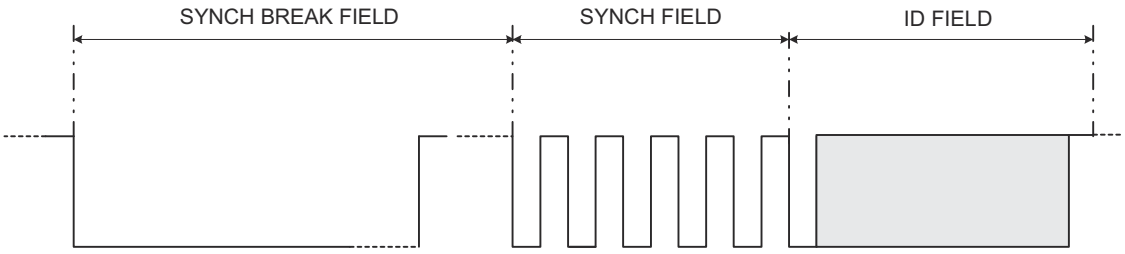


Figure 40-13. Header 3 Fields: Synch Break, Synch, and ID

40.3.1.2.2 Response

The format of the response is as illustrated in Figure 40-14. There are two types of fields in a response: data and checksum. The data field consists of exactly one data byte, one start bit, and one stop bit, for a total of 10 bits. The LSB is transmitted first. The checksum field consists of one checksum byte, one start bit and one stop bit. The checksum byte is the inverted modulo-256 sum over all data bytes in the data fields of the response.

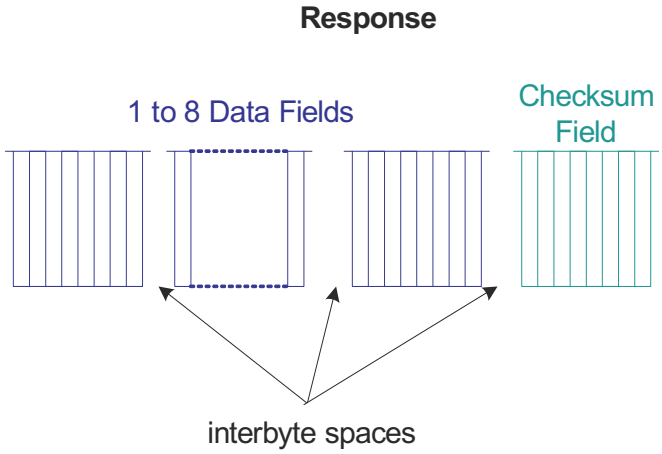


Figure 40-14. Response Format of LIN Message Frame

The format of the response is a stream of N data fields and one checksum field. Typically N is from 1 to 8, with the exception of the extended command frames (Section 40.3.1.6). The length N of the response is indicated either with the optional length control bits of the ID Field (this is used in standards earlier than LIN 1.x); see Table 40-7, or by LENGTH value in SCIFORMAT[18:16] register; see Table 40-8. The SCI/LIN module supports response lengths from 1 to 8 bytes in compliance with LIN 2.0.

Table 40-7. Response Length Info Using IDBYTE Field Bits [5:4] for LIN Standards Earlier than v1.3

ID5	ID4	Number of Data Bytes
0	0	2
0	1	2
1	0	4
1	1	8

Table 40-8. Response Length with SCIFORMAT[18:16] Programming

SCIFORMAT[18:16]	Number of Bytes
000	1
001	2
010	3
011	4
100	5
101	6
110	7
111	8

40.3.1.3 Synchronizer

The synchronizer has three major functions in the messaging between commander and responder nodes. The synchronizer generates the commander header data stream, the synchronizer synchronizes to the LIN bus for responding, and the synchronizer locally detects timeouts. A bit rate is programmed using the prescalers in the BRSR register to match the indicated LIN_speed value in the LIN description file.

The LIN synchronizer performs the following functions: commander header signal generation, responder detection and synchronization to message header with optional baud rate adjustment, response transmission timing and timeout control.

The LIN synchronizer is capable of detecting an incoming break and initializing communication at all times.

40.3.1.4 Baud Rate

The transmission baud rate of any node is configured by the CPU at the beginning; this defines the bit time T_{bit}. The bit time is derived from the fields P and M in the baud rate selection register (BRSR). There is an additional 3-bit fractional divider value, field U in the BRSR register, which further fine-tunes the data-field baud rate.

The ranges for the prescaler values in the BRSR register are:

P = 0, 1, 2, 3, . . . , 2^{24} - 1

M = 0, 1, 2, . . . , 15

U = 0, 1, 2, 3, 4, 5, 6, 7

The P, M, and U values in the BRSR register are user programmable. The P and M dividers can be used for both SCI mode and LIN mode to select a baud rate. The U value is an additional 3-bit value determining that "**a** T_{VCLK} " (with **a** = 0, 1) is added to each T_{bit} as explained in Section 40.3.1.4.2. If the ADAPT bit is set and the LIN peripheral is in adaptive baud rate mode, then all these divider values are automatically obtained during header reception when the synchronization field is measured.

The LIN protocol defines baud rate boundaries as:

$1kHz \le F_{LINCLK} \le 20kHz$

All transmitted bits are shifted in and out at T_{bit} periods.

40.3.1.4.1 Fractional Divider

The M field of the BRSR register modifies the integer prescaler P for fine tuning of the baud rate. The M value adds in increments of 1/16 of the P value.

The bit time, T_{bit} is expressed in terms of the VCLK period T_{VCLK} as follows:

For all P other than 0, and all M,

$$T_{bit} = 16\left(P + 1 + \frac{M}{16}\right)T_{VCLK}$$

For P = 0 : $T_{bit} = 32T_{VCLK}$

Therefore, the LINCLK frequency is given by:

$$F_{LINCLK} = \frac{F_{VCLK}}{16(P + 1 + \frac{M}{16})} \quad \text{For all P other than zero}$$

$$F_{LINCLK} = \frac{F_{VCLK}}{32} \quad \text{For P = 0}$$

40.3.1.4.2 Superfractional Divider

The superfractional divider scheme applies to the following modes:

- LIN commander mode (sync field + identifier field + response field + checksum field)
- LIN responder mode (response field + checksum field)

40.3.1.4.2.1 Superfractional Divider In LIN Mode

Building on the 4-bit fractional divider M (BRSR[27:24], the superfractional divider uses an additional 3-bit modulating value, illustrated in Table 40-9. The sync field (0x55), the identifier field, and the response field can all be seen as 8-bit data bytes flanked by a start bit and a stop bit. The bits with a 1 in the table have an additional VCLK period added to the T_{bit}. In LIN commander mode, bit modulation applies to sync field + identifier field + response field. In LIN responder mode, bit modulation applies to identifier field + response field.

Table 40-9. Superfractional Bit Modulation for LIN Commander Mode and Responder Mode

BRSR[30:28]	Start Bit	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]	Stop Bit
0h	0	0	0	0	0	0	0	0	0	0
1h	1	0	0	0	0	0	0	0	1	0
2h	1	0	0	0	1	0	0	0	1	0
3h	1	0	1	0	1	0	0	0	1	0
4h	1	0	1	0	1	0	1	0	1	0
5h	1	1	1	0	1	0	1	0	1	1
6h	1	1	1	0	1	1	1	0	1	1
7h	1	1	1	1	1	1	1	0	1	1

The baud rate varies over a LIN data field to average according to the BRSR[30:28] value by a *d* fraction of the peripheral internal clock: 0<d<1.

The instantaneous bit time is expressed in terms of T_{VCLK} as follows:

For all P other than 0, and all M and d (0 or 1),

$$T^{i}bit = \left[16\left(P + 1 + \frac{M}{16}\right) + d\right]T_{VCLK}$$

For P = 0, $T_{bit} = 32T_{VCLK}$

The averaged bit time is expressed in terms of T_{VCLK} as follows:

For all P other than 0, and all M and d (0 < d <1),

$$T^{a}bit = \left[16\left(P + 1 + \frac{M}{16}\right) + d\right]T_{VCLK}$$

For P = 0, $T_{bit} = 32T_{VCLK}$

40.3.1.5 Header Generation

Automatic generation of the LIN protocol header data stream is supported without CPU interaction. The CPU or the RTDMA triggers the LIN state machine to generate a message header. A commander node initiates header generation on the CPU or RTDMA writes to the IDBYTE in the LINID register. The header is always sent by the commander to initiate a LIN communication and consists of three fields: synchronization break field, synchronization field, and identification field, as seen in Figure 40-15.

Note

The LIN protocol uses the parity bits in the identifier. The control length bits are optional to the LIN protocol.

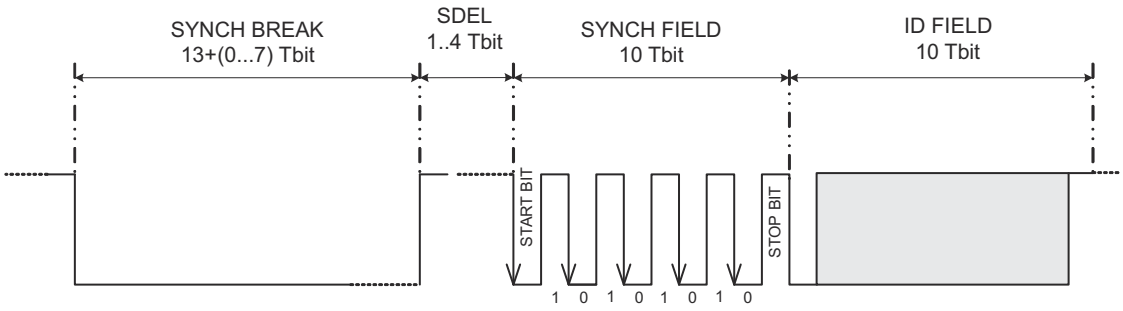


Figure 40-15. Message Header in Terms of T_{bit}

- The break field consists of two components:
  - The synchronization break (SYNCH BREAK) consists of a minimum of 13 (dominant) low bits to a maximum of 20 dominant bits. The sync break length can be extended from the minimum with the 3-bit SBREAK value in the LINCOMP register.
  - The synchronization break delimiter (SDEL) consists of a minimum of 1 (recessive) high bit to a maximum of 4 recessive bits. The delimiter marks the end of the synchronization break field. The sync break delimiter length depends on the 2-bit SDEL value in the LINCOMP register.
- The synchronization field (SYNCH FIELD) consists of one start bit, byte 0x55, and a stop bit. SYNCH FIELD is used to convey T_{bit} information and resynchronize LIN bus nodes.
- The identifier field ID byte can use 6 bits as an identifier, with optional length control and two optional bits as parity of the identifier. The identifier parity is used and checked if the PARITYENA bit is set. If length control bits are not used, then there can be a total of 64 identifiers plus parity. If neither length control or parity are used there can be up to 256 identifiers. See Figure 40-16 for an illustration of the ID field.

Note

Optional Control Length Bits

The control length bits only apply to LIN standards prior to LIN 1.3. IDBYTE field conveys response length information if compliant to standards earlier than LIN1.3. The SCIFORMAT register stores the length of the response for later versions of the LIN protocol.

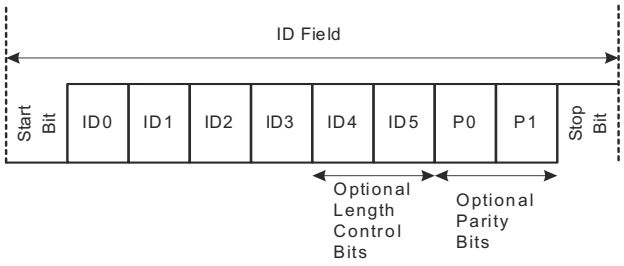


Figure 40-16. ID Field

Note

If the LIN module, configured as a responder in multibuffer mode, is in the process of transmitting data while a new header comes in, the module can end up responding with the data from the previous interrupted response (not the data corresponding to the new ID). To avoid this scenario, the following procedure can be used:

1. Check for the Bit Error (BE) during the response transmission. If the BE flag is set, this indicates that a collision has happened on the LIN bus (here because of the new Synch Break).
2. In the Bit Error ISR, configure the TD0 and TD1 registers with the next set of data to be transmitted on a TX Match for the incoming ID. Before writing to TD0/TD1 make sure that there was not already an update because of a Bit Error; otherwise, TD0/TD1 can be written twice for one ID.
3. Once the complete ID is received, based on the match, the newly configured data is transmitted by the node.

40.3.1.5.1 Event Triggered Frame Handling

The LIN 2.0 protocol uses event-triggered frames that can occasionally cause collisions. Event-triggered frames are handled in software.

If no responder answers to an event triggered frame header, the commander node sets the NRE flag, and a NRE interrupt occurs if enabled. If a collision occurs, a frame error and checksum error can arise before the NRE error. Those errors are flagged and the appropriate interrupts occur, if enabled.

Frame errors and checksum errors depend on the behavior and synchronization of the responding responders. If the responders are totally synchronized and stop transmission once the collision occurred, it is possible that only the NRE error is flagged despite the occurrence of a collision. To detect if there has been a reception of one byte before the NRE error is flagged, the BUS BUSY flag can be used as an indicator.

The BUS BUSY flag is set on the reception of the first bit of the header and remains set until the header reception is complete, and again is set on the reception of the first bit of the response. In the case of a collision, the flag is cleared in the same cycle as the NRE flag is set.

Software can implement the following sequence:

- Once the reception of the header is done (poll for RXID flag), wait for the BUS BUSY flag to get set or the NRE flag to get set.
- If the BUS BUSY flag is not set before the NRE flag, then a true no response is the case (no data has been transmitted onto the bus).
- If the BUS BUSY flag gets set, then wait for the NRE flag to get set or for successful reception. If the NRE flag is set, then a collision has occurred on the bus.

Even in the case of a collision, the received (corrupted) data is accessible in the RX buffers; registers LINRD0 and LINRD1.

40.3.1.5.2 Header Reception and Adaptive Baud Rate

A responder node baud rate can optionally be adjusted to the detected bit rate as an option to the LIN module. The adaptive baud rate option is enabled by setting the ADAPT bit. During header reception, a responder measures the baud rate during detection of the synch field. If ADAPT bit is set, then the measured baud rate is compared to the responder node programmed baud rate and adjusted to the LIN bus baud rate if necessary.

The responder node adjusts to any measured baud rate that is within ±10% of the programmed baud rate. For example, if the expected baud rate is programmed at 20kbps, the responder node detects any baud rate between 18kbps and 22kbps and adjusts accordingly. The MBRSR register prescaler is determined by the following formula:

$$\text{MBR} = \frac{\text{F}_{\text{VCLK}}}{1.1 \times \text{F}_{\text{LINCLK}}}$$

The LIN synchronizer determines two measurements: BRK_count and BAUD_count (Figure 40-17). These values are always calculated during the Header reception for synch field validation (Figure 40-18).

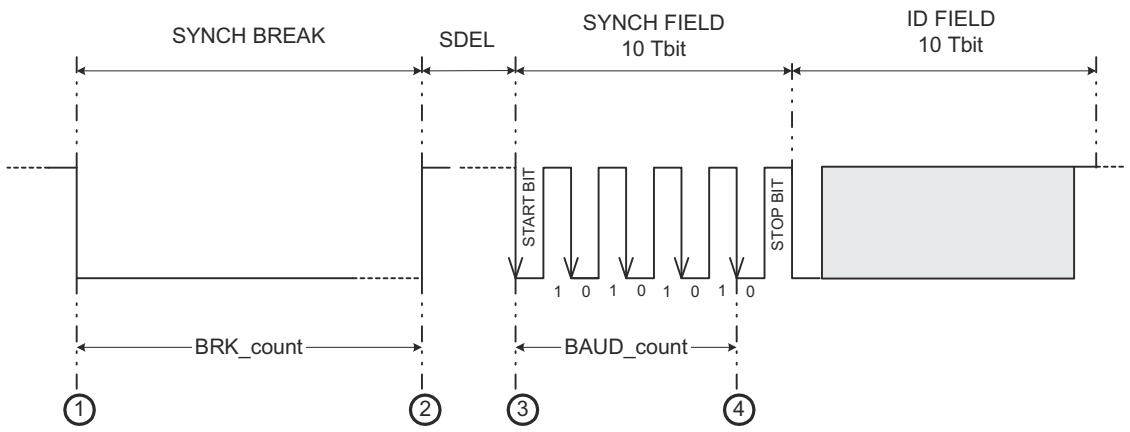


Figure 40-17. Measurements for Synchronization

By measuring the values BRK_count and BAUD_count, a valid sync break sequence can be detected as described in Figure 40-18. The four numbered events in Figure 40-17 signal the start/stop of the synchronizer counter. The synchronizer counter uses VCLK as the time base.

The synchronizer counter is used to measure the sync break relative to the detecting node T_{bit}. For a responder node receiving the sync break, a threshold of 11 T_{bit} is used as required by the LIN protocol. For detection of the dominant data stream of the sync break, the synchronizer counter is started on a falling edge and stopped on a rising edge of the LINRX. On detection of the sync break delimiter, the synchronizer counter value is saved and then reset.

On detection of five consecutive falling edges, the BAUD_count is measured. Bit timing calculation and consistency to required accuracy is implemented following the recommendations of LIN revision 2.0. A responder node can calculate a single T_{bit} time by division of BAUD_count by 8. In addition, for consistency between the detected edges the following is evaluated:

BAUD_count + BAUD_count » 2 + BAUD_count » 3 ≤ BRK_count

The BAUD_count value is shifted 3 times to the right and rounded using the first insignificant bit to obtain a T_{bit} unit. If the ADAPT bit is set, then the detected baud rate is compared to the programmed baud rate.

During the header reception processing as illustrated in Figure 40-18, if the measured BRK_count value is less than 11 T_{bit}, the sync break is not valid according to the protocol for a fixed rate. If the ADAPT bit is set, then the MBRS register is used for measuring BRK_count and BAUD_count values and automatically adjusts to any allowed LIN bus rate (refer to *LIN Specification Package 2.0*).

Note

In adaptive mode, the MBRS divider can be set to allow a maximum baud rate that is not more than 10% above the expected operating baud rate in the LIN network. Otherwise, a 0x00 data byte can mistakenly be detected as a sync break.

The break-threshold relative to the responder node is 11 T_{bit}. The break is 13 T_{bit} as specified in LIN v1.3.

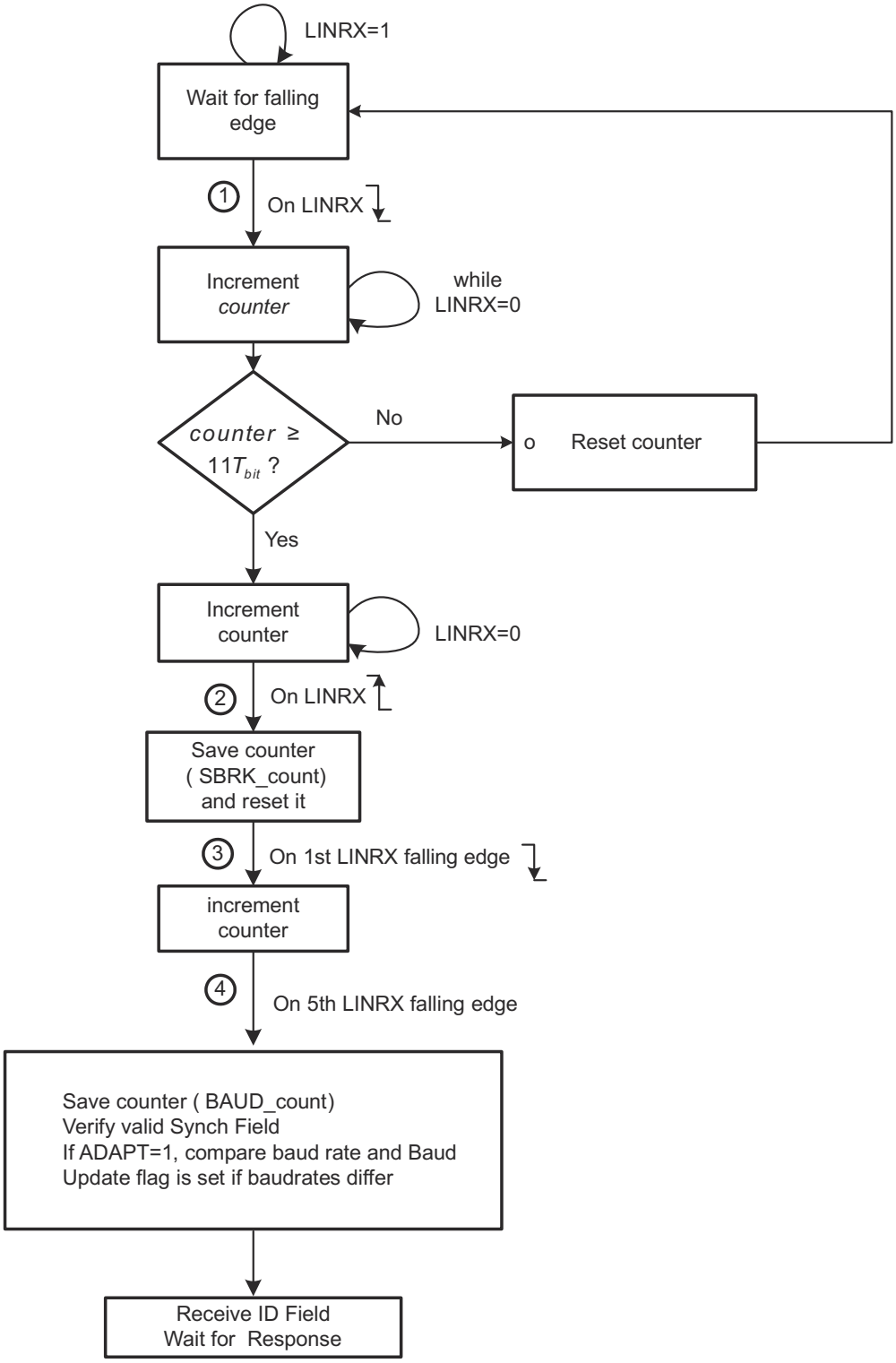


Figure 40-18. Synchronization Validation Process and Baud Rate Adjustment

If the synch field is not detected within the given tolerances, the inconsistent-sync-field-error (ISFE) flag is set. An ISFE interrupt is generated, if enabled by the respective bit in the SCISETINT register. The ID byte can be received after the synch field validation was successful. Any time a valid break (larger than 11 T_{bit}) is detected, the receiver state machine can reset to reception of this new frame. This reset condition is only valid during response state, not if an additional synch break occurs during header reception.

Note

When an inconsistent synch field (ISFE) error occurs, suggested action for the application is to reset the SWnRST bit and set the SWnRST bit to make sure that the internal state machines are back to the normal states.

40.3.1.6 Extended Frames Handling

The LIN protocol 2.0 and prior includes two extended frames with identifiers 62 (user-defined) and 63 (reserved extended). The response data length of the user-defined frame (ID 62, or 0x3E) is unlimited. The length for this identifier is set at network configuration time to be shared with the LIN bus nodes.

Extended frame communication is triggered on reception of a header with identifier 0x3E; see Figure 40-19. Once the extended frame communication is triggered, unlike normal frames, this communication needs to be stopped before issuing another header. To stop the extended frame communication the STOP EXT FRAME bit must be set.

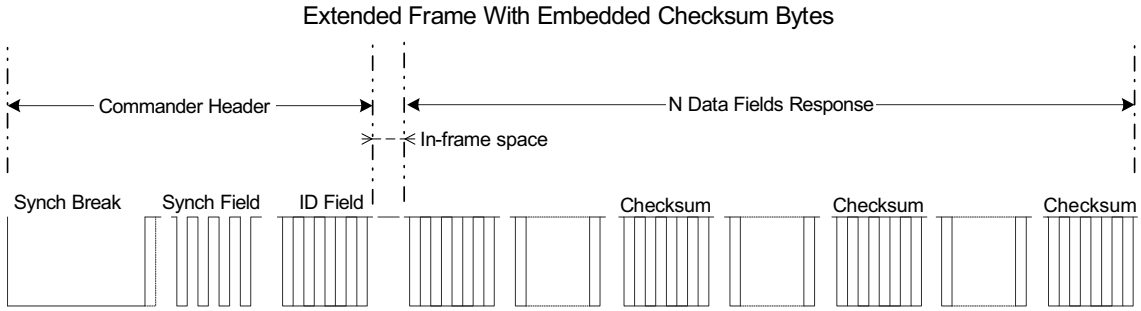


Figure 40-19. Optional Embedded Checksum in Response for Extended Frames

An ID interrupt is generated (if enabled and there is a match) on reception of ID 62 (0x3E). This interrupt allows the CPU using a software counter to keep track of the bytes that are being sent out and decides when to calculate and insert a checksum byte (recommended at periodic rates). To handle this procedure, SC bit is used. A write to the send checksum bit SC initiates an automatic send of the checksum byte. The last data field can always be a checksum in compliance with the LIN protocol.

The periodicity of the checksum insertion, defined at network configuration time, is used by the receiving node to evaluate the checksum of the ongoing message, and has the benefit of enhanced reliability.

For the sending node, the checksum is automatically embedded each time the send checksum bit SC is set. For the receiving node, the checksum is compared each time the compare checksum bit CC is set; see Figure 40-20.

Note

The LIN 2.0 enhanced checksum does not apply to the reserved identifiers. The reserved identifiers always use the classic checksum.

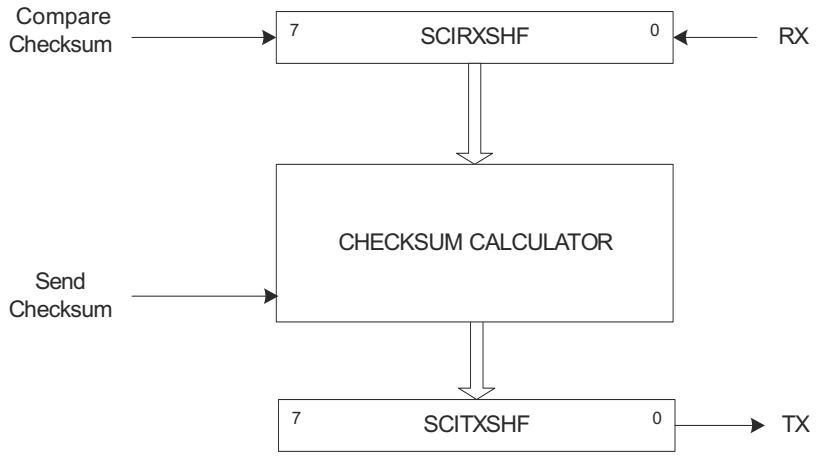


Figure 40-20. Checksum Compare and Send for Extended Frames

40.3.1.7 Timeout Control

Any LIN node listening to the bus and expecting a response initiated from a commander node can flag a no-response error timeout event. The LIN protocol defines four types of timeout events, which are all handled by the hardware of the LIN module. The four LIN protocol events are:

- No-response timeout error
- Bus idle detection
- Timeout after wakeup signal
- Timeout after three wakeup signals

40.3.1.7.1 No-Response Error (NRE)

The no-response error occurs when any node expecting a response waits for T_{FRAME_MAX} time and the message frame is not fully completed within the maximum length allowed, T_{FRAME_MAX}. After this time, a no-response error (NRE) is flagged in the NRE bit of the SCIFLR register. An interrupt is triggered, if enabled.

As specified in the LIN 1.3 standard, the minimum time to transmit a frame is:

$$T_{FRAME\_MIN} = T_{HEADER\_MIN} + T_{DATA\_FIELD} + T_{CHECKSUM\_FIELD} = 44 + 10N$$

where N = number of data fields.

And the maximum time frame is given by:

$$T_{FRAME\_MAX} = T_{FRAME\_MIN} * 1.4 = (44 + 10N) * 1.4$$

The timeout value T_{FRAME_MAX} is derived from the *N* number of data fields value, see Table 40-10. The *N* value is either embedded in the header ID field for messages or is part of the description file. In the latter case, the 3-bit CHAR value in SCIFORMAT register indicates the value for *N*.

Note

The length coding of the ID field does not apply to two extended frame identifiers, ID fields of 0x3E (62) and 0x3F (63). In these cases, the ID field can be followed by an arbitrary number of data byte fields. Also, the LIN 2.0 protocol specification mentions that ID field 0x3F (63) cannot be used. For these two cases, the NRE is not handled by the LIN hardware.

Table 40-10. Timeout Values in T_{bit} Units

N	T_{DATA_FIELD}	T_{FRAME_MIN}	T_{FRAME_MAX}
1	10	54	76
2	20	64	90
3	30	74	104
4	40	84	118
5	50	94	132
6	60	104	146
7	70	114	160
8	80	124	174

40.3.1.7.2 Bus Idle Detection

The second type of timeout can occur when a node detects an inactive LIN bus: no transitions between recessive and dominant values are detected on the bus. This happens after a minimum of 4 seconds (this is 80,000 F_{LINCLK} cycles with the fastest bus rate of 20kbps). If a node detects no activity in the bus as the TIMEOUT bit is set, assume that the LIN bus is in sleep mode. Application software can use the Timeout flag to determine when the LIN bus is inactive and put the LIN into sleep mode by writing the POWERDOWN bit.

Note

After the timeout was flagged, a SWnRESET must be asserted before entering Low-Power Mode. This is required to reset the receiver in case that an incomplete frame is on the bus before the idle period.

40.3.1.7.3 Timeout After Wakeup Signal and Timeout After Three Wakeup Signals

The third and fourth types of timeout are related to the wakeup signal. A node initiating a wakeup must expect a header from the commander within a defined amount of time: timeout after wakeup signal. See Section 40.4.3 for more details.

40.3.1.8 TXRX Error Detector (TED)

The following sources of error are detected by the TXRX error detector logic (TED). The TED logic consists of a bit monitor, an ID parity checker, and a checksum error. The following errors are detected:

- Bit errors (BE)
- Physical bus errors (PBE)
- Identifier parity errors (PE)
- Checksum errors (CE)

All of these errors (BE, PBE, PE, CE) are flagged. An interrupt for the flagged errors is generated if enabled. A message is valid for both the transmitter and the receiver if there is no error detected until the end of the frame.

40.3.1.8.1 Bit Errors

A bit error (BE) is detected at the bit time when the bit value that is monitored is different from the bit value that is sent. A bit error is indicated by the BE flag in SCIFLR. After signaling a BE, the transmission is aborted no later than the next byte. The bit monitor makes sure that the transmitted bit in LINTX is the correct value on the LIN bus by reading back on the LINRX pin as shown in Figure 40-21.

Note

If a bit occurs due to receiving a header during a responder response, NRE/TIMEOUT flag is not set for the new frame.

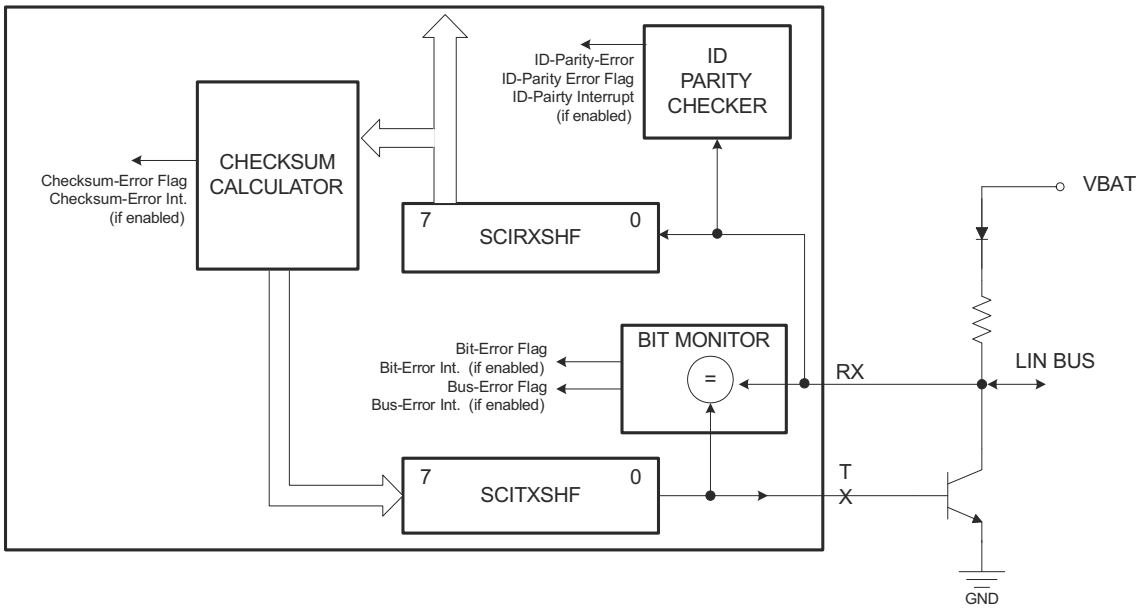


Figure 40-21. TXRX Error Detector

40.3.1.8.2 Physical Bus Errors

A Physical Bus Error (PBE) has to be detected by a commander, if no valid message can be generated on the bus (bus shorted to GND or VBAT). The bit monitor detects a PBE during the header transmission, if no Synch Break can be generated (for example, because of a bus shortage to VBAT) or if no Synch Break delimiter can be generated (for example, because of a bus shortage to GND). Once the Sync Break Delimiter was validated, all other deviations between the monitored and the sent bit value are flagged as Bit Errors (BE) for this frame.

40.3.1.8.3 ID Parity Errors

If parity is enabled, an ID parity error (PE) is detected if any of the two parity bits of the sent ID byte are not equal to the calculated parity on the receiver node. The two parity bits are generated using the following mixed parity algorithm:

$$P0 = ID0 \oplus ID1 \oplus ID2 \oplus ID4 \ (even\ Parity)$$
$$P1 = ID1 \oplus ID3 \oplus ID4 \oplus ID5 \ (odd\ Parity)$$

If an ID-parity error is detected, the ID-parity error is flagged, and the received ID is not valid. See Section 40.3.1.9 for details.

40.3.1.8.4 Checksum Errors

A checksum error (CE) is detected and flagged at the receiving end, if the calculated modulo-256 sum over all received data bytes (including the ID byte if the enhanced checksum type) plus the checksum byte does not result in 0xFF. The modulo-256 sum is calculated over each byte by adding with carry, where the carry bit of each addition is added to the LSB of the resulting sum.

For the transmitting node, the checksum byte sent at the end of a message is the inverted sum of all the data bytes (see Figure 40-22) for classic checksum implementation. The checksum byte is the inverted sum of the identifier byte and all the data bytes (see Figure 40-23) for the LIN 2.0 compliant enhanced checksum implementation. The classic checksum implementation can always be used for reserved identifiers 60 to 63; therefore, the CTYPE bit is overridden in this case. For signal-carrying-frame identifiers (0 to 59) the type of checksum used depends on the CTYPE bit.

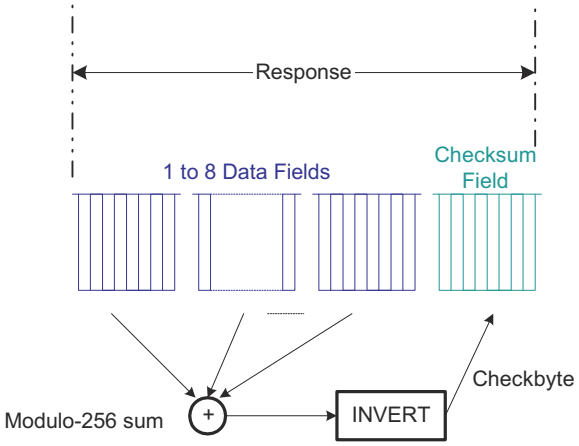


Figure 40-22. Classic Checksum Generation at Transmitting Node

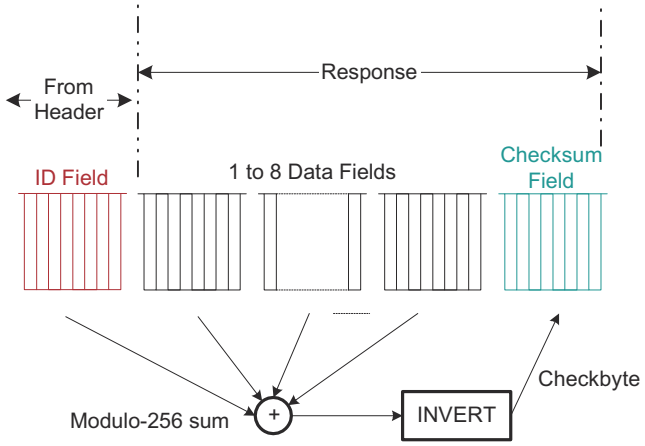


Figure 40-23. LIN 2.0-Compliant Checksum Generation at Transmitting Node

40.3.1.9 Message Filtering and Validation

Message filtering uses the entire identifier to determine which nodes participate in a response, either receiving or transmitting a response. Therefore, two acceptance masks are used as shown in Figure 40-24. During header reception, all nodes filter the ID-Field (ID-Field is the part of the header explained in Figure 40-16) to determine whether the nodes transmit a response or receive a response for the current message. There are two masks for message ID filtering: one to accept a response reception, the other to initiate a response transmission. See Figure 40-24. All nodes compare the received ID to the identifier stored in the ID-Responder Task BYTE of the LINID register and use the RX ID MASK and the TX ID MASK fields in the LINMASK register to filter the bits of the identifier that can not be compared.

If there is an RX match with no parity error and the RXENA bit is set, there is an ID RX flag and an interrupt is triggered if enabled. If there is a TX match with no parity error and the TXENA bit is set, there is an ID TX flag and an interrupt is triggered if enabled in the SCISETINT register.

The masked bits become "don't cares" for the comparison. To build a mask for a set of identifiers, an XOR function can be used.

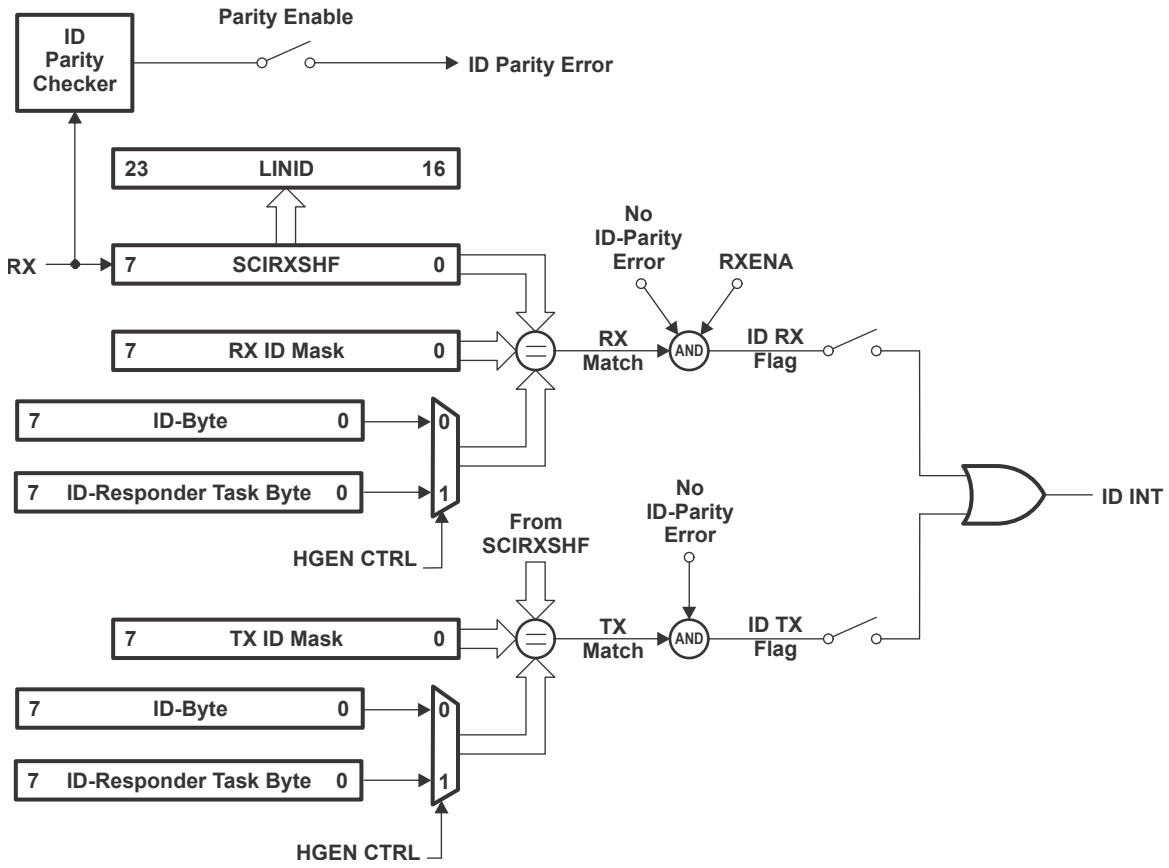


Figure 40-24. ID Reception, Filtering, and Validation

For example, to build a mask to accept IDs 0x26 and 0x25 using LINID[7:0] = 0x20; that is, compare 5 most-significant bits (MSBs) and filter 3 least-significant bits (LSBs), the acceptance mask can be:

$$(0x26 + 0x25) \oplus 0x20 = 0x07$$

A mask of all zeros compares all bits of the received identifier in the shift register with the ID-BYTE in LINID[7:0]. If HGEN CTRL is set to 1, a mask of 0xFF always causes a match. A mask of all 1s filters all bits of the received identifier, and thus there is an ID match regardless of the content of the ID-Responder Task BYTE field in the LINID register.

Note

When the HGEN CTRL bit = 0, the LIN nodes compare the received ID to the ID-BYTE field in the LINID register, and use the RX ID MASK and the TX ID MASK in the LINMASK register to filter the bits of the identifier that can not be compared.

If there is an RX match with no parity error and the RXENA bit is set, there is an ID RX flag and an interrupt is triggered if enabled. A mask of all 0s compares all bits of the received identifier in the shift register with the ID-BYTE field in LINID[7:0]. A mask of all 1s filters all bits of the received identifier and there is no match.

If HGEN CTLR = 1:

- Received ID is compared with the ID-Responder Task byte, using the RXID mask and the TXID mask.
- A mask of all 1s always result in a match.
- A mask of all 0s means all the bits must be the same to result in a match.
- If a mask has some bits that are 1s, then those bits are not used for the filtering criterion.

If HGEN CTRL = 0:

- Received ID is compared with the ID byte, using the RXID mask and the TXID mask.
- A mask of all 1s results in no match.
- A mask of all 0s means all the bits must be the same to result in a match.
- If a mask has some bits that are 1s, then those bits are not used for the filtering criterion.

During header reception, the received identifier is copied to the Received ID field LINID[23:16]. If there is no parity error and there is either a TX match or an RX match, then the corresponding TX or RX ID flag is set. If the ID interrupt is enabled, then an ID interrupt is generated.

After the ID interrupt is generated, the CPU can read the Received ID field LINID[23:16] and determine what response to load into the transmit buffers.

Note

When byte 0 is written to TD0 (LINTD0[31:24]), the response transmission is automatically generated.

In multibuffer mode, the TXRDY flag is set when all the response data bytes and checksum byte are copied to the shift register SCITXSHF. In non-multibuffer mode, the TXRDY flag is set each time a byte is copied to the SCITXSHF register, and also for the last byte of the frame after the checksum byte is copied to the SCITXSHF register.

In multibuffer mode, the TXEMPTY flag is set when both the transmit buffers TDy and the SCITXSHF shift register are emptied and the checksum has been sent. In non-multibuffer mode, TXEMPTY is set each time TD0 and SCITXSHF are emptied, except for the last byte of the frame where the checksum byte must also be transmitted.

If parity is enabled, all responder receiving nodes validate the identifier using all eight bits of the received ID byte. The SCI/LIN flags a corrupted identifier if an ID-parity error is detected.

40.3.1.10 Receive Buffers

To reduce CPU load when receiving a LIN N-byte (with N = 1–8) response in interrupt mode or RTDMA mode, the SCI/LIN module has eight receive buffers. These buffers can store an entire LIN response in the RDy receive buffers. Figure 40-8 illustrates the receive buffers.

The checksum byte following the data bytes is validated by the internal checksum calculator. The checksum error (CE) flag indicates a checksum error and a CE interrupt is generated if enabled in the SCISETINT register.

The multibuffer 3-bit counter counts the data bytes transferred from the SCIRXSHF register to the RDy receive buffers if multibuffer mode is enabled, or to RD0 if multibuffer mode is disabled. The 3-bit compare register contains the number of data bytes expected to be received. In cases where the IDBYTE field does not convey message length (see *Note: Optional Control Length Bits* in Section 40.3.1.5), the LENGTH value, indicates the expected length and is used to load the 3-bit compare register. Whether the length control field or the LENGTH value is used is selectable with the COMMMODE bit.

A receive interrupt, and a receive ready RXRDY flag , and a RTDMA request (RXDMA) can occur after receiving a response, if there are no response receive errors for the frame (such as, there is no checksum error, frame error, and overrun error). The checksum byte is compared before acknowledging a reception. A RTDMA request can be generated for each received byte or for the entire response depending on whether the multibuffer mode is enabled or not (MBUFMODE bit).

Note

In multibuffer mode following are the scenarios associated with clearing the RXRDY flag bit:

1. The RXRDY flag cannot be cleared by reading the corresponding interrupt offset in the SCIINTVECT0/1 register.
2. For LENGTH less than or equal to 4, Read to RD0 register clears the RXRDY flag.
3. For LENGTH greater than 4, Read to RD1 register clears the RXRDY flag.

40.3.1.11 Transmit Buffers

To reduce the CPU load when transmitting a LIN N-byte (with N = 1–8) response in interrupt modeor RTDMA mode, the SCI/LIN module has 8 transmit buffers, TD0–TD7 in LINTD0 and LINTD1. With these transmit buffers, an entire LIN response field can be preloaded in the TDy transmit buffers. Optionally, a RTDMA transfer can be done on a byte-per-byte basis when multibuffer mode is not enabled (MBUFMODE bit). Figure 40-9 illustrates the transmit buffers.

The multibuffer 3-bit counter counts the data bytes transferred from the TDy transmit buffers register if multibuffer mode is enabled, or from TD0 to SCITXSHF if multibuffer mode is disabled. The 3-bit compare register contains the number of data bytes expected to be transmitted. If the ID field is not used to convey message length (see *Note: Optional Control Length Bits* in Section 40.3.1.5), the LENGTH value indicates the expected length and is used instead to load the 3-bit compare register. Whether the length control field or the LENGTH value is used is selectable with the COMMMODE bit.

A transmit interrupt (TX interrupt) and a transmit ready flag (TXRDY flag), as well as a RTDMA request (TXDMA) can occur after transmitting a response.A RTDMA request can be generated for each transmitted byte or for the entire response depending on whether multibuffer mode is enabled or not (MBUFMODE bit).

The checksum byte is automatically generated by the checksum calculator and sent after the data-fields transmission is finished. The multibuffer 3-bit counter counts the data bytes transferred from the TDy buffers into the SCITXSHF register.

Note

The transmit interrupt request can be eliminated until the next series of data is written into the transmit buffers LINTD0 and LINTD1, by disabling the corresponding interrupt using the SCICLRINT register or by disabling the transmitter using the TXENA bit.

40.3.2 LIN Interrupts

LIN and SCI modes have a common interrupt block, as explained in Section 40.2.2. There are 16 interrupt sources in the SCI/LIN module, with 8 of them being LIN mode only, as seen in Table 40-4.

A LIN message frame indicating the timing and sequence of the LIN interrupts that can occur is shown in Figure 40-25.

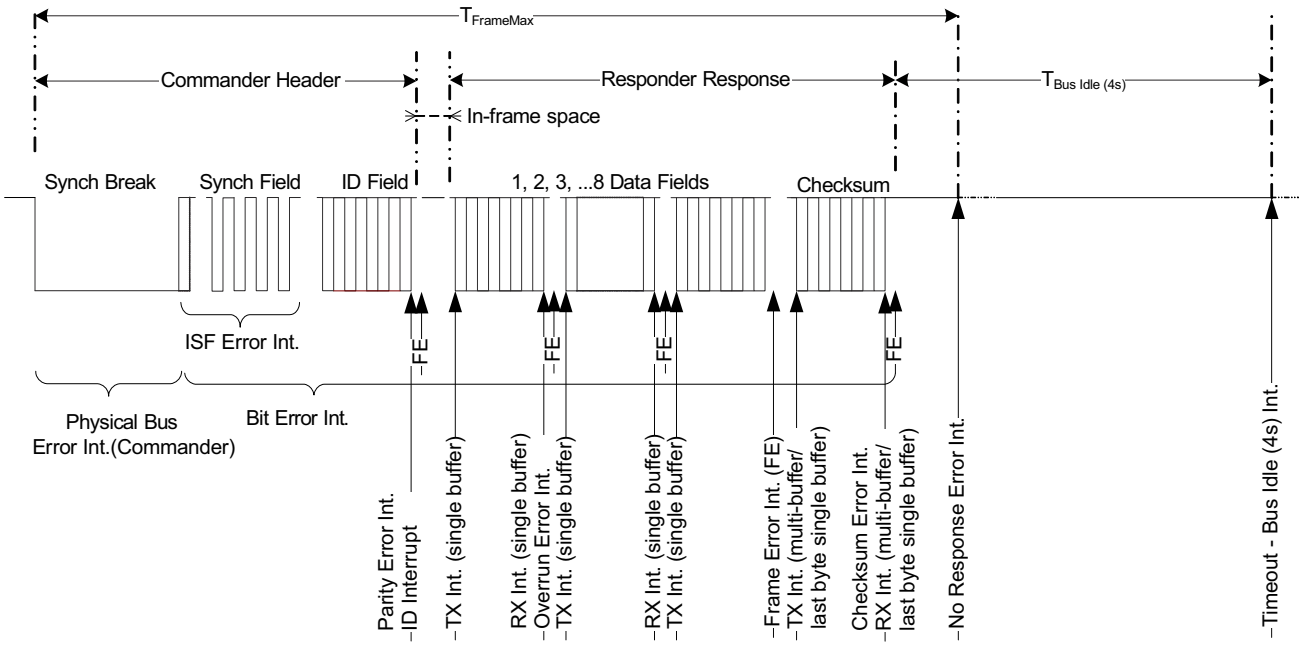


Figure 40-25. LIN Message Frame Showing LIN Interrupt Timing and Sequence

40.3.3 Servicing LIN Interrupts

When servicing an interrupt, clear the corresponding flag in the flag register (SCIFLR) before clearing the global interrupt (LIN_GLB_INT_CLR). The ISR can follow the guidelines below. This prevents any spurious or duplicate interrupt from occurring.

- Clear the LIN interrupt flag in the SCIFLR register.
- Read the LIN interrupt status register to make sure the flag is cleared.
- Clear the global interrupt flag bit in LIN_GLB_INT_CLR.

Note

The transmit interrupt is generated before the LIN transmitter is ready to accept new data. Inside of the LIN transmit ISR, the software can wait until the buffer is completely empty before loading the next data. This can be done by polling for the Bus Busy Flag (SCIFLR.BUSY) to be 0.

40.3.4 LIN RTDMA Interface

The LIN RTDMA interface uses the SCI RTDMA interface logic. RTDMA requests for receive (RXDMA request) and transmit (TXDMA request) are available for the SCI/LIN module. There are two modes for RTDMA transfers depending on whether multibuffer mode is enabled or not using the multibuffer enable control bit (MBUFMODE).

Note

Do not use the RTDMA to transmit data to multiple peripheral IDs. Writing to the LINID register initiates a new transmission. The RTDMA writes to the LINID register before the LIN state machine is ready to accept the new ID. Doing so causes the LIN to miss this transmission.

40.3.4.1 LIN Receive RTDMA Requests

In LIN mode, when the multibuffer option is enabled, if a received response (up to eight data bytes) is transferred to the receive buffers (RDy), then a RTDMA request is generated. If the multibuffer option is disabled, then RTDMA requests are generated on a byte-per-byte basis until all the expected response data fields are received. This RTDMA functionality is enabled and disabled using the SET_RX_DMA and CLRRXDMA bits, respectively.

40.3.4.2 LIN Transmit RTDMA Requests

In LIN mode with the multibuffer option enabled, after a transmission (up to eight data bytes stored in the transmit buffers (TDy) in the LINTD0 and LINTD1 registers), a RTDMA request is generated to reload the transmit buffer for the next transmission. If the multibuffer option is disabled, then RTDMA requests are generated on a byte-per-byte basis until all bytes are transferred. This RTDMA functionality is enabled and disabled using the SET_TX_DMA and CLRTXDMA bits, respectively.

40.3.5 LIN Configurations

The following list details the configuration steps that software can perform prior to the transmission or reception of data in LIN mode. As long as the SWnRST bit in the SCIGCR1 register is cleared to 0 the entire time that the LIN is being configured, the order in which the registers are programmed is not important.

- Enable LIN by setting RESET bit (SCIGCR0.0).
- Clear SWnRST to 0 before configuring the LIN (SCIGCR1.7).
- Enable the LINRX and LINTX pins by setting the RXFUNC and TXFUNC bits.
- Select LIN mode by programming the LINMODE bit (SCIGCR1.6).
- Select commander or responder mode by programming the CLOCK bit.
- Select the desired frame format (checksum, parity, length control) by programming SCIGCR1.
- Select multibuffer mode by programming MBUFMODE bit (SCIGCR1.10).
- Select the baud rate to be used for communication by programming BRSR.
- Set the maximum baud rate to be used for communication by programming MBRSR.
- Set the CONT bit to make LIN not halt for an emulation breakpoint until the LIN current reception or transmission is complete (this bit is used only in an emulation environment).
- Set LOOPBACK bit (SCIGCR1.16) to connect the transmitter to the receiver internally if needed (this feature is used to perform a self-test).
- Select the receiver enable RXENA bit (SCIGCR1.24), if data is to be received.
- Select the transmit enable TXENA bit (SCIGCR1.25), if data is to be transmitted.
- Select the RXIDMASK and the TXIDMASK fields in the LINMASK register.
- Set SWnRST (SCIGCR1.7) to 1 after the LIN is configured.
- Receive or Transmit data (see Section 40.3.1.9, Section 40.3.5.1, and Section 40.3.5.2).

Note

If TXENA is set and the SWnRST is released, the LIN immediately generates a new RTDMA request but not a new transmit interrupt request. If using interrupts, the first transmission must be started with software by writing data to the transmit buffer, followed by writing the chosen ID to the LINID register to initiate the transmission.

40.3.5.1 Receiving Data

The LIN receiver is enabled to receive messages if both the RXFUNC bit and the RXENA bit are set to 1. If the RXFUNC bit is not set, the LINRX pin functions as a general-purpose I/O pin rather than as a LIN function pin.

The IDRXFLAG in the SCIFLR register is set after a valid LINID is received with an RX Match. An ID interrupt is then generated, if enabled.

40.3.5.1.1 Receiving Data in Single-Buffer Mode

Single-buffer mode is selected when the MBUFMODE bit is cleared to 0. In this mode, LIN sets the RXRDY bit when the LIN transfers newly received data from SCIRXSHF to RD0. The SCI clears the RXRDY bit after the new data in RD0 has been read. Also, as data is transferred from SCIRXSHF to RD0, the LIN sets the FE, OE, or PE flags if any of these error conditions were detected in the received data. These error conditions are supported with configurable interrupt capability.

You can receive data by:

1. Polling Receive Ready Flag
2. Receive Interrupt
3. RTDMA

In polling method, software can poll for the RXRDY bit and read the data from RD0 byte of the LINRD0 register once the RXRDY bit is set high. The CPU is unnecessarily overloaded by selecting the polling method. To avoid this, you can use the interrupt or RTDMA method. To use the interrupt method, the SETRXINT bit is set. To use the RTDMA method, the SET_RX_DMA bit must be set. Either an interrupt or a RTDMA request is generated the moment the RXRDY bit is set. If the checksum scheme is enabled by setting the Compare Checksum (CC) bit to 1, the checksum is compared on the byte that is currently being received, which is expected to be the checksum byte. The CC bit is cleared once the checksum is received. A CE is immediately flagged, if there is a checksum error.

40.3.5.1.2 Receiving Data in Multibuffer Mode

Multibuffer mode is selected when the MBUFMODE bit is set to 1. In this mode, LIN sets the RXRDY bit after receiving the programmed number of data in the receive buffer and the checksum field, the complete frame. The error condition detection logic is similar to the single-buffer mode, except that this logic monitors for the complete frame. Like single-buffer mode, you can use the polling, RTDMA, or interrupt method to read the data. The received data has to be read from the LINRD0 and LINRD1 registers, based on the number of bytes. For a LENGTH less than or equal to 4, a read from the LINRD0 register clears the RXRDY flag. For a LENGTH greater than 4, a read from the LINRD1 register clears the RXRDY flag. If the checksum scheme is enabled by setting the Compare Checksum (CC) bit to 1 during the reception of the data, then the byte that is received after the reception of the programmed number of data bytes indicated by the LENGTH field is treated as a checksum byte. The CC bit is cleared once the checksum is received and compared.

40.3.5.2 Transmitting Data

The LIN transmitter is enabled if both the TXFUNC bit and the TXENA bit are set to 1. If the TXFUNC bit is not set, the LINTX pin functions as a general-purpose I/O pin rather than as a LIN function pin. Any value written to the TD0 before the TXENA bit is set to 1 is not transmitted. Both of these control bits allow for the LIN transmitter to be held inactive independently of the receiver.

The IDTXFLAG bit in the SCIFLR register is set after a valid LIN ID is received with a TX Match. An ID interrupt is then generated, if enabled.

40.3.5.2.1 Transmitting Data in Single-Buffer Mode

Single-buffer mode is selected when the MBUFMODE bit is cleared to 0. In this mode, LIN waits for data to be written to TD0, transfers the data to SCITXSHF, and transmits the data. The TXRDY and TXEMPTY bits indicate the status of the transmit buffers. That is, when the transmitter is ready for data to be written to TD0, the TXRDY bit is set. Additionally, if both TD0 and SCITXSHF are empty, then the TXEMPTY bit is also set.

You can transmit data by:

1. Polling Transmit Ready Flag
2. Transmit Interrupt
3. RTDMA

In polling method, software can poll for the TXRDY bit to go high before writing the data to the TD0. The CPU is unnecessarily overloaded by selecting the polling method. To avoid this, you can use the interrupt or RTDMA method. To use the interrupt method, the SETXINT bit is set. To use the RTDMA method, the SET_TX_DMA bit is set. Either an interrupt or a RTDMA request is generated the moment the TXRDY bit is set. When the LIN has completed transmission of all pending frames, the SCITXSHF register and the TD0 are empty, the TXRDY bit is set, and an interrupt/RTDMA request is generated, if enabled. Because all data has been transmitted, the interrupt/RTDMA request can be halted. This can either be done by disabling the transmit interrupt (CLRTXINT)/ RTDMA request (CLRTXDMA bit) or by disabling the transmitter (clear TXENA bit). If the checksum scheme is enabled by setting the Send Checksum (SC) bit to 1, the checksum byte is sent after the current byte transmission. The SC bit is cleared after the checksum byte has been transmitted.

Note

The TXRDY flag cannot be cleared by reading the corresponding interrupt offset in the SCIINTVECT0 or SCIINTVECT1 register.

40.3.5.2.2 Transmitting Data in Multibuffer Mode

Multibuffer mode is selected when the MBUFMODE bit is set to 1. Like single-buffer mode, you can use the polling, RTDMA, or interrupt method to write the data to be transmitted. The transmitted data has to be written to the LINTD0 and LINTD1 registers, based on the number of bytes. LIN waits for data to be written to Byte 0 (TD0) of the LINTD0 register and transfers the programmed number of bytes to SCITXSHF to transmit one by one automatically. If the checksum scheme is enabled by setting the Send Checksum (SC) bit to 1, the checksum is sent after transmission of the last byte of the programmed number of data bytes, indicated by the LENGTH field. The SC bit is cleared after the checksum byte has been transmitted.

40.4 Low-Power Mode

The SCI/LIN module can be put in either local or global low-power mode. Global low-power mode is asserted by the system and is not controlled by the SCI/LIN module. During global low-power mode, all clocks to the SCI/LIN are turned off so the module is completely inactive. If global low-power mode is requested while the receiver is receiving data, then the SCI/LIN completes the current reception and then enters the low-power mode, that is, module enters low-power mode only when Busy bit (SCIFLR.3) is cleared.

The LIN module can enter low-power mode either when there was no activity on the LINRX pin for more than 4 seconds (this can be either a constant recessive or dominant level) or when a Sleep Command frame was received. Once the Timeout flag (SCIFLR.4) was set or once a Sleep Command was received, the POWERDOWN bit (SCIGCR2.0) must be set by the application software to make the module enter local low-power mode. A wakeup signal terminates the sleep mode of the LIN bus.

Note

Enabling Local Low-Power Mode During Receive and Transmit

If the wakeup interrupt is enabled and low-power mode is requested while the receiver is receiving data, then the SCI/LIN immediately generates a wakeup interrupt to clear the power-down bit. Thus, the SCI/LIN is prevented from entering low-power mode and completes the current reception. Otherwise, if the wakeup interrupt is disabled, the SCI/LIN completes the current reception and then enters the low-power mode.

40.4.1 Entering Sleep Mode

In LIN protocol, a sleep command is used to broadcast the sleep mode to all nodes. The sleep command consists of a diagnostic commander request frame with identifier 0x3C (60), with the first data field as 0x00. There must be no activity in the bus once all nodes receive the sleep command: the bus is in sleep mode.

Local low-power mode is asserted by setting the POWERDOWN bit; setting this bit stops the clocks to the SCI/LIN internal logic and registers. Clearing the POWERDOWN bit causes SCI/LIN to exit from local low-power mode. All the registers are accessible during local power-down mode. If a register is accessed in low-power mode, this access results in enabling the clock to the module for that particular access alone.

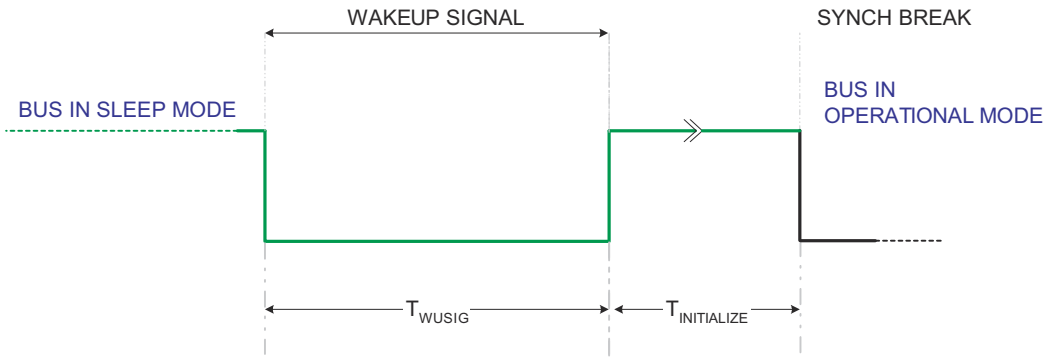
40.4.2 Wakeup

The wakeup interrupt is used to allow the SCI/LIN module to automatically exit a low-power mode. A SCI/LIN wakeup is triggered when a low level is detected on the receive RX pin, and this clears the POWERDOWN bit.

Note

If the wakeup interrupt is disabled, then the SCI/LIN enters low-power mode whenever the SCI/LIN is requested to do so, but a low level on the receive RX pin does not cause the SCI/LIN to exit low-power mode.

In LIN mode, any node can terminate sleep mode by sending a wakeup signal, see Figure 40-26. A responder node that detects the bus in sleep mode, and with a wakeup request pending, sends a wakeup signal. The wakeup signal is a dominant value on the LIN bus for T_{WUSIG}; this is at least 5 T_{bits} for the LIN bus baud rates. The wakeup signal is generated by sending a 0xF0 byte containing 5 dominant T_{bits} and 5 recessive T_{bits}.



0.25ms ≤ T_{WUSIG} ≤ 5ms

Figure 40-26. Wakeup Signal Generation

Assuming a bus with no noise or loading effects, a write of 0xF0 to TD0 loads the transmitter to meet the wakeup signal timing requirement for T_{WUSIG}. Then, setting the GENWU bit transmits the preloaded value in TD0 for a wakeup signal transmission.

Note

The GENWU bit can be set/reset only when SWnRST is set to 1 and the node is in power-down mode. The bit is cleared on a valid synch break detection. A commander sending a wakeup request, exits power-down mode upon reception of the wakeup pulse. The bit is cleared on a SWnRST. This can be used to stop a commander from sending further wakeup requests.

The TI TPIC1021 LIN transceiver, upon receiving a wakeup signal, translates it to the microcontroller for wakeup with a dominant level on the RX pin, or a signal to the voltage regulator. While the POWERDOWN bit is set, if the LIN module detects a recessive-to-dominant edge (falling edge) on the RX pin, the LIN module generates a wakeup interrupt if enabled in the SCISETINT register.

According to LIN protocol 2.0, the TI TPIC1021 LIN transceiver detecting a dominant level on the bus longer than 150ms detects it as a wakeup request. The LIN responder is ready to listen to the bus in less than 100ms ($T_{INITIALIZE}$<100ms) after a dominant-to-recessive edge (end-of-wakeup signal).

40.4.3 Wakeup Timeouts

The LIN protocol defines the following timeouts for a wakeup sequence. After a wakeup signal has been sent to the bus, all nodes wait for the commander to send a header. If no synch field is detected before 150ms (3,000 cycles at 20kHz) after a wakeup signal is transmitted, a new wakeup is sent by the same node that requested the first wakeup. This sequence is not repeated more than two times. After three attempts to wake up the LIN bus, wakeup signal generation is suspended for a 1.5s (30,000 cycles at 20kHz) period after three breaks.

Note

To achieve compatibility to LIN1.3 timeout conditions, the MBRS register must be set to make sure that the LIN 2.0 (real-time-based) timings meet the LIN 1.3 bit time base. A node triggering the wakeup can set the MBRS register accordingly to meet the targeted time as 128 Tbits × programmed prescaler.

The LIN handles the wakeup expiration times defined by the LIN protocol with a hardware implementation.

40.5 Emulation Mode

In emulation mode, the CONT bit determines how the SCI/LIN operates when the program is suspended. The SCI/LIN counters are affected by this bit during debug mode. when set, the counters are not stopped and when cleared, the counters are stopped debug mode.

Any reads in emulation mode to a SCI/LIN register do not have any effect on the flags in the SCIFLR register.

Note

When emulation mode is entered during the Frame transmission or reception of the frame and CONT bit is not set, Communication is not expected to be successful. The suggested usage is to set CONT bit during emulation mode for successful communication.

40.6 Software

40.6.1 LIN Registers to Driverlib Functions

Table 40-11. LIN Registers to Driverlib Functions

File	Driverlib Function
SCIGCR0	
lin.h	LIN_enableModule
lin.h	LIN_disableModule
SCIGCR1	
lin.h	LIN_setLINMode
lin.h	LIN_setMessageFiltering
lin.h	LIN_enableParity
lin.h	LIN_disableParity
lin.h	LIN_setCommMode
lin.h	LIN_enableAutomaticBaudrate
lin.h	LIN_disableAutomaticBaudrate
lin.h	LIN_stopExtendedFrame
lin.h	LIN_setChecksumType
lin.h	LIN_enableSCIMode
lin.h	LIN_disableSCIMode
lin.h	LIN_setSCICommMode
lin.h	LIN_enableSCIParity
lin.h	LIN_disableSCIParity
lin.h	LIN_setSCIStopBits
lin.h	LIN_enableSCISleepMode
lin.h	LIN_disableSCISleepMode
lin.h	LIN_enterSCILowPower
lin.h	LIN_exitSCILowPower
lin.h	LIN_setSCICharLength
lin.h	LIN_setSCIFrameLength
lin.h	LIN_isSCIDataAvailable
lin.h	LIN_isSCISpaceAvailable
lin.h	LIN_readSCICharNonBlocking
lin.h	LIN_readSCICharBlocking
lin.h	LIN_writeSCICharNonBlocking
lin.h	LIN_writeSCICharBlocking
lin.h	LIN_enableSCIModuleErrors
lin.h	LIN_disableSCIModuleErrors
lin.h	LIN_enableSCIInterrupt
lin.h	LIN_disableSCIInterrupt
lin.h	LIN_clearSCIInterruptStatus
lin.h	LIN_setSCIInterruptLevel0
lin.h	LIN_setSCIInterruptLevel1
lin.h	LIN_isSCIReceiverIdle
lin.h	LIN_getSCITxFrameType
lin.h	LIN_getSCIRxFrameType
lin.h	LIN_isSCIBreakDetected

Table 40-11. LIN Registers to Driverlib Functions (continued)

File	Driverlib Function
lin.h	LIN_enableDataTransmitter
lin.h	LIN_disableDataTransmitter
lin.h	LIN_enableDataReceiver
lin.h	LIN_disableDataReceiver
lin.h	LIN_performSoftwareReset
lin.h	LIN_enterSoftwareReset
lin.h	LIN_exitSoftwareReset
lin.h	LIN_enableIntLoopback
lin.h	LIN_disableIntLoopback
lin.h	LIN_enableMultibufferMode
lin.h	LIN_disableMultibufferMode
lin.h	LIN_setDebugSuspendMode
SCIGCR2	
lin.h	LIN_sendWakeupSignal
lin.h	LIN_enterSleep
lin.h	LIN_sendChecksum
lin.h	LIN_triggerChecksumCompare
lin.h	LIN_enterSCILowPower
lin.h	LIN_exitSCILowPower
SCISETINT	
lin.h	LIN_enableInterrupt
lin.h	LIN_setInterruptLevel1
lin.h	LIN_enableSCIInterrupt
lin.h	LIN_setSCIInterruptLevel1
lin.h	LIN_getInterruptLevel
SCICLEARINT	
lin.h	LIN_disableInterrupt
lin.h	LIN_setInterruptLevel0
lin.h	LIN_disableSCIInterrupt
lin.h	LIN_setSCIInterruptLevel0
SCISETINTLVL	
lin.h	LIN_setInterruptLevel1
lin.h	LIN_setSCIInterruptLevel1
lin.h	LIN_getInterruptLevel
SCICLEARINTLVL	
lin.h	LIN_setInterruptLevel0
lin.h	LIN_setSCIInterruptLevel0
SCIFLR	
lin.h	LIN_isTxReady
lin.h	LIN_isRxReady
lin.h	LIN_isTxMatch
lin.h	LIN_isRxMatch
lin.h	LIN_clearInterruptStatus
lin.h	LIN_isSCIDataAvailable
lin.h	LIN_isSCISpaceAvailable

Table 40-11. LIN Registers to Driverlib Functions (continued)

File	Driverlib Function
lin.h	LIN_clearSCIInterruptStatus
lin.h	LIN_isSCIReceiverIdle
lin.h	LIN_getSCITxFrameType
lin.h	LIN_getSCIRxFrameType
lin.h	LIN_isSCIBreakDetected
lin.h	LIN_isBusBusy
lin.h	LIN_isTxBufferEmpty
lin.h	LIN_getInterruptStatus
SCIINTVECT0	
lin.h	LIN_getInterruptLine0Offset
SCIINTVECT1	
lin.h	LIN_getInterruptLine1Offset
SCIFORMAT	
lin.c	LIN_sendData
lin.c	LIN_getData
lin.h	LIN_setFrameLength
lin.h	LIN_setSCICharLength
lin.h	LIN_setSCIFrameLength
BRSR	
lin.h	LIN_setBaudRatePrescaler
SCIED	
lin.h	LIN_readSCICharNonBlocking
lin.h	LIN_readSCICharBlocking
SCIRD	
-	
SCITD	
lin.h	LIN_writeSCICharNonBlocking
lin.h	LIN_writeSCICharBlocking
SCIPIO0	
lin.h	LIN_enableModule
lin.h	LIN_disableModule
SCIPIO2	
lin.h	LIN_getPinStatus
COMP	
lin.h	LIN_setSyncFields
RD0	
lin.c	LIN_getData
RD1	
-	
MASK	
lin.h	LIN_setTxMask
lin.h	LIN_setRxMask
lin.h	LIN_getTxMask
lin.h	LIN_getRxMask
ID	

Table 40-11. LIN Registers to Driverlib Functions (continued)

File	Driverlib Function
lin.h	LIN_setIDByte
lin.h	LIN_setIDResponderTask
lin.h	LIN_getRxIdentifier
TD0	
lin.c	LIN_sendData
lin.h	LIN_sendWakeupSignal
TD1	
-	
MBRSR	
lin.h	LIN_setMaximumBaudRate
IODFTCTRL	
lin.h	LIN_enableModuleErrors
lin.h	LIN_disableModuleErrors
lin.h	LIN_enableSCIModuleErrors
lin.h	LIN_disableSCIModuleErrors
lin.h	LIN_enableExtLoopback
lin.h	LIN_disableExtLoopback
lin.h	LIN_setTransmitDelay
lin.h	LIN_setPinSampleMask
GLB_INT_EN	
lin.h	LIN_enableGlobalInterrupt
lin.h	LIN_disableGlobalInterrupt
GLB_INT_FLG	
lin.h	LIN_getGlobalInterruptStatus
GLB_INT_CLR	
lin.h	LIN_clearGlobalInterruptStatus

40.6.2 LIN Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/lin

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

40.6.2.1 LIN Internal Loopback with Interrupts - SINGLE_CORE

FILE: lin_ex1_loopback_interrupt.c

This example configures the LIN module in commander mode for internal loopback with interrupts. The module is setup to perform 8 data transmissions with different transmit IDs and varying transmit data. Upon reception of an ID header, an interrupt is triggered on line 0 and an interrupt service routine (ISR) is called. The received data is then checked for accuracy.

External Connections

- None.

Watch Variables

- txData - An array with the data being sent
- rxData - An array with the data that was received
- result - The example completion status (PASS = 0xABCD, FAIL = 0xFFFF)

- level0Count - The number of line 0 interrupts
- level1Count - The number of line 1 interrupts

40.6.2.2 LIN SCI Mode Internal Loopback with Interrupts - SINGLE_CORE

FILE: lin_ex2_sci_loopback.c

This example configures the LIN module in SCI mode for internal loopback with interrupts. The LIN module performs as a SCI with a set character and frame length in a non-multi-buffer mode. The module is setup to continuously transmit a character, wait to receive that character, and repeat.

External Connections

- None.

Watch Variables

- rxCount - The number of RX interrupts
- transmitChar - The character being transmitted
- receivedChar - The character received

40.6.2.3 LIN SCI MODE Internal Loopback with DMA - SINGLE_CORE

FILE: lin_ex3_sci_dma.c

This example configures the LIN module in SCI mode for internal loopback with the use of the DMA. The LIN module performs as SCI with a set character and frame length in multi-buffer mode. When the transmit buffers in the LINTD0 and LINTD1 registers have enough space, the DMA will transfer data from global variable sData into those transmit registers. Once the received buffers in the LINRD0 and LINRD1 registers contain data,the DMA will transfer the data into the global variable rdata.

When all data has been placed into rData, a check of the validity of the data will be performed in one of the DMA channels' ISRs.

External Connections

- None

Watch Variables

- *sData* - Data to send
- *rData* - Received data

40.6.2.4 LIN Internal Loopback without interrupts (polled mode) - SINGLE_CORE

FILE: lin_ex4_loopback_polling.c

This example configures the LIN module in commander mode for internal loopback without interrupts. The module is setup to perform 8 data transmissions with different transmit IDs and varying transmit data. Waits for reception of an ID header. The received data is then checked for accuracy.

External Connections

- None.

Watch Variables

- txData - An array with the data being sent
- rxData - An array with the data that was received
- result - The example completion status (PASS = 0xABCD, FAIL = 0xFFFF)

40.6.2.5 LIN SCI MODE (Single Buffer) Internal Loopback with DMA - SINGLE_CORE

FILE: lin_ex5_sci_dma_single_buffer.c

This example configures the LIN module in SCI mode for internal loopback with the use of the DMA. The LIN module performs as SCI with a set character and frame length in single-buffer compatibility mode. When the transmit buffer i.e. the SCITD register is free, the DMA will transfer data from global variable sData into this

register. Once the received buffer, i.e. the SCIRD register contains data, the DMA will transfer the data into the global variable rdata.

When all data has been placed into rData, a check of the validity of the data will be performed in one of the DMA channels' ISRs.

External Connections

- None

Watch Variables

- *sData* - Data to send
- *rData* - Received data

40.7 LIN Registers

This Section describes the LIN Registers.

40.7.1 LIN Base Address Table

Table 40-12. LIN Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
LIN_REGS	LINA_BASE	0x6006_0000	YES	YES	YES	YES	YES	YES	-	YES
LIN_REGS	LINB_BASE	0x6006_1000	YES	YES	YES	YES	YES	YES	-	YES

40.7.2 LIN_REGS Registers

Table 40-13 lists the memory-mapped registers for the LIN_REGS registers. All register offset addresses not listed in Table 40-13 should be considered as reserved locations and the register contents should not be modified.

Table 40-13. LIN_REGS Registers

Offset	Acronym	Register Name	Protection
0h	SCIGCR0	Global Control Register 0	
4h	SCIGCR1	Global Control Register 1	
8h	SCIGCR2	Global Control Register 2	
Ch	SCISETINT	Interrupt Enable Register	
10h	SCICLEARINT	Interrupt Disable Register	
14h	SCISETINTLVL	Set Interrupt Level Register	
18h	SCICLEARINTLVL	Clear Interrupt Level Register	
1Ch	SCIFLR	Flag Register	
20h	SCIINTVECT0	Interrupt Vector Offset Register 0	
24h	SCIINTVECT1	Interrupt Vector Offset Register 1	
28h	SCIFORMAT	Length Control Register	
2Ch	BRSR	Baud Rate Selection Register	
30h	SCIED	Emulation buffer Register	
34h	SCIRD	Receiver data buffer Register	
38h	SCITD	Transmit data buffer Register	
3Ch	SCIPIO0	Pin control Register 0	
44h	SCIPIO2	Pin control Register 2	
60h	LINCOMP	Compare register	
64h	LINRD0	Receive data register 0	
68h	LINRD1	Receive data register 1	
6Ch	LINMASK	Acceptance mask register	
70h	LINID	LIN ID Register	
74h	LINTD0	Transmit Data Register 0	
78h	LINTD1	Transmit Data Register 1	
7Ch	MBRSR	Maximum Baud Rate Selection Register	
90h	IODFTCTRL	IODFT for LIN	
E0h	LIN_GLB_INT_EN	LIN Global Interrupt Enable Register	
E4h	LIN_GLB_INT_FLG	LIN Global Interrupt Flag Register	
E8h	LIN_GLB_INT_CLR	LIN Global Interrupt Clear Register	

Complex bit access types are encoded to fit into small table cells. Table 40-14 shows the codes that are used for access types in this section.

Table 40-14. LIN_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear

Table 40-14. LIN_REGS Access Type Codes (continued)

Access Type	Code	Description
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

40.7.2.1 SCIGCR0 Register (Offset = 0h) [Reset = 00000000h]

SCIGCR0 is shown in Figure 40-27 and described in Table 40-15.

Return to the Summary Table.

The SCIGCR0 register defines the module reset.

Figure 40-27. SCIGCR0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							RESET
R-0h							R/W-0h

Table 40-15. SCIGCR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	RESET	R/W	0h	This bit resets the SCI/LIN module. This bit is effective in LIN or SCI-compatible mode.. This bit affects the reset state of the SCI/LIN module. Reset type: SYSRSn 0h (R/W) = SCI/LIN module is in held in reset. 1h (R/W) = SCI/LIN module is out of reset.

40.7.2.2 SCIGCR1 Register (Offset = 4h) [Reset = 00000000h]

SCIGCR1 is shown in Figure 40-28 and described in Table 40-16.

Return to the Summary Table.

The SCIGCR1 register defines the frame format, protocol, and communication mode used by the SCI.

Figure 40-28. SCIGCR1 Register

31	30	29	28	27	26	25	24
RESERVED						TXENA	RXENA
R-0h						R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED						CONT	LOOPBACK
R-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		STOPEXTFRAME	HGENCTRL	CTYPE	MBUFMODE	ADAPT	SLEEP
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SWnRST	LINMODE	CLK_COMMANDER	STOP	PARITY	PARITYENA	TIMINGMODE	COMMMODE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 40-16. SCIGCR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	TXENA	R/W	0h	Transmit enable. This bit is effective in LIN and SCI modes. Data is transferred from SCITD or the TDy (with y=0, 1,...7) buffers in LIN mode to the SCITXSHF shift out register only when the TXENA bit is set. Note: Data written to SCITD or the transmit multi-buffer before TXENA is set is not transmitted. If TXENA is cleared while transmission is ongoing, the data previously written to SCITD is sent (including the checksum byte in LIN mode). Reset type: SYSRSn 0h (R/W) = Disable transfers from SCITD or TDy to SCITXSHF 1h (R/W) = Enable transfers of data from SCITD or TDy to SCITXSHF

Table 40-16. SCIGCR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	RXENA	R/W	0h	Receive enable. This bit is effective in LIN or SCI-compatible mode. RXENA allows or prevents the transfer of data from SCIRXSHF to SCIRD or the receive multibuffers. Note: Clearing RXENA stops received characters from being transferred into the receive buffer or multi-buffers, prevents the RX status flags (see Table 7) from being updated by receive data, and inhibits both receive and error interrupts. However, the shift register continues to assemble data regardless of the state of RXENA. Note: If RXENA is cleared before the time the reception of a frame is complete, the data from the frame is not transferred into the receive buffer. Note: If RXENA is set before the time the reception of a frame is complete, the data from the frame is transferred into the receive buffer. If RXENA is set while SCIRXSHF is in the process of assembling a frame, the status flags are not guaranteed to be accurate for that frame. To ensure that the status flags correctly reflect what was detected on the bus during a particular frame, RXENA should be set before the detection of that frame Reset type: SYSRSn 0h (R/W) = Prevents the receiver from transferring data from the shift buffer to the receive buffer or multi-buffers 1h (R/W) = Allows the receiver to transfer data from the shift buffer to the receive buffer or multi-buffers
23-18	RESERVED	R	0h	Reserved
17	CONT	R/W	0h	Continue on suspend. This bit has an effect only when a program is being debugged with an emulator, and it determines how the SCI/LIN operates when the program is suspended. This bit affects the LIN counters. When this bit is set, the counters are not stopped during debug. When this bit is cleared, the counters are stopped during debug. Reset type: SYSRSn 0h (R/W) = When debug mode is entered, the SCI/LIN state machine is frozen. Transmissions and LIN counters are halted and resume when debug mode is exited. 1h (R/W) = When debug mode is entered, the SCI/LIN continues to operate until the current transmit and receive functions are complete.
16	LOOPBACK	R/W	0h	Loopback bit. This bit is effective in LIN or SCI-compatible mode. The self-checking option for the SCI/LIN can be selected with this bit. If the LINTX and LINRX pins are configured with SCI/LIN functionality, then the LINTX pin is internally connected to the LINRX pin. Externally, during loop back operation, the LINTX pin outputs a high value and the LINRX pin is in a high-impedance state. If this bit value is changed while the SCI/LIN is transmitting or receiving data, errors may result. Reset type: SYSRSn 0h (R/W) = Loopback mode is disabled. 1h (R/W) = Loopback mode is enabled.
15-14	RESERVED	R	0h	Reserved
13	STOPEXTFRAME	R/W	0h	Stop extended frame communication. This bit is effective in LIN mode only. This bit can be written only during extended frame communication. When the extended frame communication is stopped, this bit is cleared automatically. Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Extended frame communication will be stopped, once current frame transmission/reception is completed.

Table 40-16. SCIGCR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	HGENCTRL	R/W	0h	HGEN control bit. This bit is effective in LIN mode only. This bit controls the type of mask filtering comparison. Reset type: SYSRSn 0h (R/W) = ID filtering using ID-Byte. RECEIVEDID and IDBYTE fields in the LINID register are used for detecting a match (using TX/RXMASK values). Mask of 0xFF in LINMASK register will result in NO match. 1h (R/W) = ID filtering using ID-RESPONDERTask byte (Recommended). RECEIVEDID and IDRESPONDERTASKBYTE fields in the LINID register are used for detecting a match (using TX/RXMASK values). Mask of 0xFF in LINMASK register will result in ALWAYS match
11	CTYPE	R/W	0h	Checksum type. This bit is effective in LIN mode only. This bit controls the type of checksum to be used: classic or enhanced. Reset type: SYSRSn 0h (R/W) = Classic checksum is used. This checksum is compatible with LIN 1.3 Responder nodes. The classic checksum contains the modulo-256 sum with carry over all data bytes. Frames sent with Identifier 60 (0x3C) to 63 (0x3F) must allways use the classic checksum. 1h (R/W) = Enhanced checksum is used. The enhanced checksum is compatible with LIN 2.0 and newer Responder nodes. The enhanced checksum contains the modulo-256 sum with carry over all data bytes AND the protected Identifier.
10	MBUFMODE	R/W	0h	Multibuffer mode. This bit is effective in LIN or SCI-compatible mode. This bit controls receive/transmit buffer usage, that is, whether the RX/TX multibuffers are used or a single register, RD0/TD0, is used. Reset type: SYSRSn 0h (R/W) = The multi-buffer mode is disabled. 1h (R/W) = The multi-buffer mode is enabled.
9	ADAPT	R/W	0h	Adapt mode enable. This mode is effective in LIN mode only. This bit has an effect during the detection of the Sync Field. There are two LIN protocol bit rate modes that could be enabled with this bit according to the Node capability file definition: automatic or select. Software and network configuration will decide which of the previous two modes. When this bit is cleared, the LIN 2.0 protocol fixed bit rate should be used. If the ADAPT bit is set, a LIN Responder node detecting the baudrate will compare it to the prescalers in BRSR register and update it if they are different. The BRSR register will be updated with the new value. If this bit is not set there will be no adjustment to the BRSR register. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Automatic baudrate adjustment is disabled. 1h (R/W) = Automatic baudrate adjustment is enabled.

Table 40-16. SCIGCR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	SLEEP	R/W	0h	SCI sleep. SCI compatibility mode only. In a multiprocessor configuration, this bit controls the receive sleep function. Clearing this bit brings the SCI out of sleep mode. The receiver still operates when the SLEEP bit is set however, RXRDY is updated and SCIRD is loaded with new data only when an address frame is detected. The remaining receiver status flags are updated and an error interrupt is requested if the corresponding interrupt enable bit is set, regardless of the value of the SLEEP bit. In this way, if an error is detected on the receive data line while the SCI is asleep, software can promptly deal with the error condition. The SLEEP bit is not automatically cleared when an address byte is detected. This field is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = Sleep mode is disabled. 1h (R/W) = Sleep mode is enabled.
7	SWnRST	R/W	0h	Software reset (active low). This bit is effective in LIN or SCI-compatible mode. The SCI/LIN should only be configured while SWnRST = 0. Only the following configuration bits can be changed in runtime (i.e., while SWnRESET = 1): - STOP EXT Frame (SCIGCR1[13]) - CC bit (SCIGCR2[17]) - SC bit (SCIGCR2[16]) Reset type: SYSRSn 0h (R/W) = The SCI/LIN is in its reset state no data will be transmitted or received. Writing a 0 to this bit intializes the SCI/LIN state machines and operating flags. All affected logic is held in the reset state until a 1 is written to this bit. 1h (R/W) = The SCI/LIN is in its ready state transmission and reception can occur. After this bit is set to 1, the configuration of the module should not change.
6	LINMODE	R/W	0h	LIN mode This bit controls the mode of operation of the module. Reset type: SYSRSn 0h (R/W) = LIN mode is disabled SCI compatibility mode is enabled. 1h (R/W) = LIN mode is enabled SCI compatibility mode is disabled.
5	CLK_COMMANDER	R/W	0h	SCI internal clock enable or LIN Commander/Responder configuration. In the SCI mode, this bit enables the clock to the SCI module. In LIN mode, this bit determines whether a LIN node is a Responder or Commander. Reset type: SYSRSn 0h (R/W) = SCI-compatible mode: Reserved. LIN mode: The module is in Responder mode. 1h (R/W) = SCI-compatible mode: Enable clock to the SCI module. LIN mode: The node is in Commander mode.
4	STOP	R/W	0h	SCI number of stop bits. This bit is effective in SCI-compatible mode only. Note: The receiver checks for only one stop bit. However in idle-line mode, the receiver waits until the end of the second stop bit (if STOP = 1) to begin checking for an idle period. This field is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = One stop bit is used. 1h (R/W) = Two stop bits are used.

Table 40-16. SCIGCR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	PARITY	R/W	0h	SCI parity odd/even selection. This bit is effective in SCI-compatible mode only. If the PARITY ENA bit (SCIGCR1.2) is set, PARITY designates odd or even parity. The parity bit is calculated based on the data bits in each frame and the address bit (in address-bit mode). The start and stop fields in the frame are not included in the parity calculation. This field is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = Odd parity is used. The SCI transmits and expects to receive a value in the parity bit that makes odd the total number of bits in the frame with the value of 1. 1h (R/W) = Even parity is used. The SCI transmits and expects to receive a value in the parity bit that makes even the total number of bits in the frame with the value of 1.
2	PARITYENA	R/W	0h	Parity enable. Enables or disables the parity function. Reset type: SYSRSn 0h (R/W) = SCI-compatible mode: Parity disabled no parity bit is generated during transmission or is expected during reception. LIN mode: ID-parity verification is disabled. 1h (R/W) = SCI compatible mode: Parity enabled. A parity bit is generated during transmission and is expected during reception. LIN mode: ID-parity verification is enabled.
1	TIMINGMODE	R/W	0h	SCI timing mode bit. This bit is effective in SCI-compatible mode only. It must be set to 1 when the SCI mode is used. This bit configures the SCI for asynchronous operation. Reset type: SYSRSn 0h (R/W) = Reserved. 1h (R/W) = Must be set to 1 when module is configured for SCI operation
0	COMMMODE	R/W	0h	SCI/LIN communication mode bit. In compatibility mode, it selects the SCI communication mode. In LIN mode it selects length control option for ID-field bits ID4 and ID5. Reset type: SYSRSn 0h (R/W) = SCI-compatible mode: Idle-line mode is used. LIN mode: ID4 and ID5 are not used for length control. 1h (R/W) = SCI-compatible mode: Address-bit mode is used. LIN mode: ID4 and ID5 are used for length control.

40.7.2.3 SCIGCR2 Register (Offset = 8h) [Reset = 00000000h]

SCIGCR2 is shown in Figure 40-29 and described in Table 40-17.

Return to the Summary Table.

The SCIGCR2 register is used to send or compare a checksum byte during extended frames, to generate a wakeup and for low-power mode control of the LIN module.

Figure 40-29. SCIGCR2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						CC	SC
R-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							GENWU
R-0h							R/W-0h
7	6	5	4	3	2	1	0
RESERVED							POWERDOWN
R-0h							R/W-0h

Table 40-17. SCIGCR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17	CC	R/W	0h	Compare Checksum. This mode is effective in LIN mode only. This bit is used by the receiver for extended frames to trigger a checksum compare. The user will initiate this transaction by writing a one to this bit. In non multibuffer mode, once the CC bit is set, the checksum will be compared on the byte that is currently being received, expected to be the checkbyte. During Multi-buffer mode, following are the scenarios associated with the CC bit : - If CC bit is set during the reception of the data, then the byte that is received after the reception of the programmed no. of data bytes indicated by SCIFORMAT[18:16], is treated as a checksum byte. - If CC bit is set during the IDLE period (i.e. during inter-frame space), then the next immediate byte will be treated as a checksum byte. A CE will immediatly be flagged if there is a checksum error. This bit is automatically cleared once the checksum is successfully compared. Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Compare checksum on expected checkbyte

Table 40-17. SCIGCR2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	SC	R/W	0h	Send Checksum This mode is effective in LIN mode only. This bit is used by the transmitter with extended frames to send a checkbyte. In non multibuffer mode the checkbyte will be sent after the current byte transmission. In multibuffer mode the checkbyte will be sent after the last byte count, indicated by the SCIFORMAT[18:16]). This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No checkbyte will be sent. 1h (R/W) = A checkbyte will be sent. This bit will automatically get cleared after the checkbyte is transmitted. The checksum will not be sent if this bit is set before transmitting the very first byte, that is, during interframe space.
15-9	RESERVED	R	0h	Reserved
8	GENWU	R/W	0h	Generate wakeup signal. This bit controls the generation of a wakeup signal, by transmitting the TDO buffer value. This bit is cleared on reception of a valid sync break. Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Transmit TDO for wakeup. This bit will be cleared on a SWnRST (SCIGCR1.7)
7-1	RESERVED	R	0h	Reserved
0	POWERDOWN	R/W	0h	Power down. This bit is effective in LIN or SCI-compatible mode. When the powerdown bit is set, the SCI/LIN module attempts to enter local low-power mode. If the POWERDOWN bit is set while the receiver is actively receiving data and the wakeup interrupt is disabled, then the SCI/LIN will delay low-power mode from being entered until completion of reception. In LIN mode the user may set the POWERDOWN bit on Sleep Command reception or on idle bus detection (more than 4 seconds, i.e. 80,000 cycles at 20kHz) Reset type: SYSRSn 0h (R/W) = Normal operation 1h (R/W) = Request local low-power mode

40.7.2.4 SCISETINT Register (Offset = Ch) [Reset = 00000000h]

SCISETINT is shown in Figure 40-30 and described in Table 40-18.

Return to the Summary Table.

The SCISETINT register is used to enable the various interrupts available in the LIN module.

Figure 40-30. SCISETINT Register

31	30	29	28	27	26	25	24
SETBEINT	SETPBEINT	SETCEINT	SETISFEINT	SETNREINT	SETFEINT	SETOEINT	SETPEINT
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
23	22	21	20	19	18	17	16
RESERVED					SET_RX_DMA_ALL	SET_RX_DMA	SET_TX_DMA
R-0h					R/W1S-0h	R/W1S-0h	R/W1S-0h
15	14	13	12	11	10	9	8
RESERVED		SETIDINT	RESERVED			SETRXINT	SETTXINT
R-0h		R/W1S-0h	R-0h			R/W1S-0h	R/W1S-0h
7	6	5	4	3	2	1	0
SETTOA3WUSINT	SETTOAWUSINT	RESERVED	SETTIMEOUTINT	RESERVED		SETWAKEUPINT	SETBRKDTINT
R/W1S-0h	R/W1S-0h	R-0h	R/W1S-0h	R-0h		R/W1S-0h	R/W1S-0h

Table 40-18. SCISETINT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SETBEINT	R/W1S	0h	Set bit error interrupt. This bit is effective in LIN mode only. Setting this bit enables the SCI/LIN module to generate an interrupt when there is a bit error. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
30	SETPBEINT	R/W1S	0h	Set physical bus error interrupt. This bit is effective in LIN mode only. Setting this bit enables the SCI/LIN module to generate an interrupt when a physical bus error occurs. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
29	SETCEINT	R/W1S	0h	Set checksum-error Interrupt. This bit is effective in LIN mode only. Setting this bit enables the SCI/LIN module to generate an interrupt when there is a checksum error. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
28	SETISFEINT	R/W1S	0h	Set inconsistent-sync-field-error interrupt. This bit is effective in LIN mode only. Setting this bit enables the SCI/LIN module to generate an interrupt when there is an inconsistent sync field error. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.

Table 40-18. SCISETINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	SETNREINT	R/W1S	0h	Set no-response-error interrupt. This bit is effective in LIN mode only. Setting this bit enables the SCI/LIN module to generate an interrupt when a no-response error occurs. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
26	SETFEINT	R/W1S	0h	Set framing-error interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit enables the SCI/LIN module to generate an interrupt when a framing error occurs. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
25	SETOEINT	R/W1S	0h	Set overrun-error interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit enables the SCI/LIN module to generate an interrupt when an overrun error occurs. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
24	SETPEINT	R/W1S	0h	Set parity interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit enables the SCI/LIN module to generate an interrupt when a parity error occurs. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
23-19	RESERVED	R	0h	Reserved
18	SET_RX_DMA_ALL	R/W1S	0h	Set receiver DMA for Address & Data frames. This bit is effective in LIN or SCI-compatible mode for devices with a DMA module. To enable RX DMA request for address and data frames this bit must be set. If it is cleared, RX interrupt request is generated for address frames and DMA requests are generated for data frames. Reset type: SYSRSn 0h (R/W) = Receiver DMA request is disabled for address frames (RX interrupt request is enabled for address frames). Writing a 0 to this bit has no effect. 1h (R/W) = Receiver DMA request is enabled for address and data frames
17	SET_RX_DMA	R/W1S	0h	Set receiver DMA. This bit is effective in LIN or SCI-compatible mode for devices with a DMA module. To enable DMA requests for the receiver this bit must be set. If it is cleared, interrupt requests are generated depending on SETRXINT. Reset type: SYSRSn 0h (R/W) = Receiver DMA request is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Receiver DMA request is enabled.
16	SET_TX_DMA	R/W1S	0h	Set transmit DMA. This bit is effective in LIN or SCI-compatible mode for devices with a DMA module. To enable DMA requests for the transmitter, this bit must be set. If it is cleared, interrupt requests are generated depending on SETTXINT. Reset type: SYSRSn 0h (R/W) = Transmit DMA request is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Transmit DMA request is enabled

Table 40-18. SCISETINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-14	RESERVED	R	0h	Reserved
13	SETIDINT	R/W1S	0h	Set Identification interrupt. This bit is effective in LIN mode only. This bit is set to enable interrupt once a valid matching identifier is received. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
12-10	RESERVED	R	0h	Reserved
9	SETRXINT	R/W1S	0h	Set Receiver interrupt. Setting this bit enables the SCI/LIN to generate a receive interrupt after a frame has been completely received and the data is being transferred from SCIRXSHF to SCIRD. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
8	SETTXINT	R/W1S	0h	Set Transmitter interrupt. Setting this bit enables the SCI/LIN to generate a transmit interrupt as data is being transferred from SCITD to SCITXSHF and the TXRDY bit is being set. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
7	SETTOA3WUSINT	R/W1S	0h	Set Timeout After 3 Wakeup Signals interrupt. This bit is effective in LIN mode only. Setting this bit enables the SCI/LIN to generate an interrupt when there is a timeout after 3 wakeup signals have been sent. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
6	SETTOAWUSINT	R/W1S	0h	Set Timeout After Wakeup Signal interrupt. This bit is effective in LIN mode only. Setting this bit enables the SCI/LIN to generate an interrupt when there is a timeout after one wakeup signal has been sent. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
5	RESERVED	R	0h	Reserved
4	SETTIMEOUTINT	R/W1S	0h	Set timeout interrupt. This bit is effective in LIN mode only. Setting this bit enables the SCI/LIN to generate an interrupt when no LIN bus activity (bus idle) occurs for at least 4 seconds. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
3-2	RESERVED	R	0h	Reserved

Table 40-18. SCISETINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	SETWAKEUPINT	R/W1S	0h	Set wake-up interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit enables the SCI/LIN to generate a wake-up interrupt and thereby exit low-power mode. The wake-up interrupt is asserted on falling edge of the wake-up pulse. If enabled, the wake-up interrupt is asserted when local low-power mode is requested while the receiver is busy or if a low level is detected on the SCIRX pin during low-power mode. Wake-up interrupt is not asserted upon a wakeup pulse if the module is not in power down mode. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
0	SETBRKDTINT	R/W1S	0h	Set break-detect interrupt. This bit is effective in SCI-compatible mode only. Setting this bit enables the SCI/LIN to generate an interrupt if a break condition is detected on the LINRX pin. This field is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.

40.7.2.5 SCICLEARINT Register (Offset = 10h) [Reset = 00000000h]

SCICLEARINT is shown in Figure 40-31 and described in Table 40-19.

Return to the Summary Table.

The SCICLEARINT register is used to disable the enabled interrupts without accessing the SCISETINT register.

Figure 40-31. SCICLEARINT Register

31	30	29	28	27	26	25	24
CLRBEINT	CLRPBEINT	CLRCEINT	CLRISFEINT	CLRNREINT	CLRFEINT	CLROEINT	CLRPEINT
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	16
RESERVED					RESERVED	CLRRXDMA	CLRTXDMA
R-0h					R-0h	R/W1C-0h	R/W1C-0h
15	14	13	12	11	10	9	8
RESERVED		CLRIDINT	RESERVED			CLRRXINT	CLRTXINT
R-0h		R/W1C-0h	R-0h			R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
CLRTOA3WUSINT	CLRTOAWUSINT	RESERVED	CLRTIMEOUTINT	RESERVED		CLRWAKEUPINT	CLRBRKDTINT
R/W1C-0h	R/W1C-0h	R-0h	R/W1C-0h	R-0h		R/W1C-0h	R/W1C-0h

Table 40-19. SCICLEARINT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CLRBEINT	R/W1C	0h	Clear Bit Error Interrupt. This bit is effective in LIN mode only. Setting this bit disables the bit error interrupt. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
30	CLRPBEINT	R/W1C	0h	Clear Physical Bus Error Interrupt. This bit is effective in LIN mode only. Setting this bit disables the physical-bus error interrupt. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
29	CLRCEINT	R/W1C	0h	Clear checksum-error Interrupt. This bit is effective in LIN mode only. Setting this bit disables the checksum-error interrupt. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
28	CLRISFEINT	R/W1C	0h	Clear Inconsistent-Sync-Field-Error Interrupt. This bit is effective in LIN mode only. Setting this bit disables the ISFE interrupt. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.

Table 40-19. SCICLEARINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	CLRNREINT	R/W1C	0h	Clear No-Reponse-Error Interrupt. This bit is effective in LIN mode only. Setting this bit disables the no-response error interrupt. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
26	CLRFEINT	R/W1C	0h	Clear Framing-Error Interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit disables framing-error interrupt. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
25	CLROEINT	R/W1C	0h	Clear Overrun-Error Interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit disables the overrun interrupt. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
24	CLRPEINT	R/W1C	0h	Clear Parity Interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit disables the parity error interrupt. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
23-19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	CLRRXDMA	R/W1C	0h	Clear receiver DMA. This bit is effective in LIN or SCI-compatible mode. Setting this bit disables the receive DMA request. Reset type: SYSRSn 0h (R/W) = Receiver DMA request is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Receiver DMA request is enabled. Writing a 1 to this bit will disable the DMA request and clear this bit.
16	CLRTXDMA	R/W1C	0h	Clear transmit DMA. This bit is effective in LIN or SCI-compatible mode. Setting this bit disables the transmit DMA request. Reset type: SYSRSn 0h (R/W) = Transmit DMA request is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Transmit DMA request is enabled. Writing a 1 to this bit will disable the DMA request and clear this bit.
15-14	RESERVED	R	0h	Reserved
13	CLRIDINT	R/W1C	0h	Clear Identifier interrupt. This bit is effective in LIN mode only. Setting this bit disables the ID interrupt. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
12-10	RESERVED	R	0h	Reserved

Table 40-19. SCICLEARINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	CLRRXINT	R/W1C	0h	Clear Receiver interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit disables the receiver interrupt. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
8	CLRTXINT	R/W1C	0h	Clear Transmitter interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit disables the transmitter interrupt. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
7	CLRTOA3WUSINT	R/W1C	0h	Clear Timeout After 3 Wakeup Signals interrupt. This bit is effective in LIN mode only. Setting this bit disables the timeout after 3 wakeup signals interrupt. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
6	CLRTOAWUSINT	R/W1C	0h	Clear Timeout After Wakeup Signal interrupt. This bit is effective in LIN mode only. Setting this bit disables the timeout after one wakeup signal interrupt. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
5	RESERVED	R	0h	Reserved
4	CLRTIMEOUTINT	R/W1C	0h	Clear Timeout interrupt. This bit is effective in LIN mode only. Setting this bit disables the timeout (LIN bus idle) interrupt. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
3-2	RESERVED	R	0h	Reserved
1	CLRWAKEUPINT	R/W1C	0h	Clear Wake-up interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit disables the wake-up interrupt. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
0	CLRBRKDTINT	R/W1C	0h	Clear Break-detect interrupt. This bit is effective in SCI-compatible mode only. Setting this bit disables the Break-detect interrupt. This field is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.

40.7.2.6 SCISETINTLVL Register (Offset = 14h) [Reset = 00000000h]

SCISETINTLVL is shown in Figure 40-32 and described in Table 40-20.

Return to the Summary Table.

The SCISETINTLVL register is used to map individual interrupt sources to the INT1 interrupt line.

Figure 40-32. SCISETINTLVL Register

31	30	29	28	27	26	25	24
SETBEINTLVL	SETPBEINTLVL	SETCEINTLVL	SETISFEINTLVL	SETNREINTLVL	SETFEINTLVL	SETOEINTLVL	SETPEINTLVL
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
23	22	21	20	19	18	17	16
RESERVED					RESERVED	RESERVED	
R-0h					R-0h	R-0h	
15	14	13	12	11	10	9	8
RESERVED		SETIDINTLVL	RESERVED			SETRXINTOVO	SETTXINTLVL
R-0h		R/W1S-0h	R-0h			R/W1S-0h	R/W1S-0h
7	6	5	4	3	2	1	0
SETTOA3WUSINTLVL	SETTOAWUSINTLVL	RESERVED	SETTIMEOUTINTLVL	RESERVED		SETWAKEUPINTLVL	SETBRKDTINTLVL
R/W1S-0h	R/W1S-0h	R-0h	R/W1S-0h	R-0h		R/W1S-0h	R/W1S-0h

Table 40-20. SCISETINTLVL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SETBEINTLVL	R/W1S	0h	Set Bit Error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the Bit Error interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
30	SETPBEINTLVL	R/W1S	0h	Set Physical Bus Error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the Physical Bus Error interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
29	SETCEINTLVL	R/W1S	0h	Set Checksum-error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the Checksum-error interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
28	SETISFEINTLVL	R/W1S	0h	Set Inconsistent-Sync-Field-Error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the Inconsistent-Sync-Field-Error interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.

Table 40-20. SCISETINTLVL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	SETNREINTLVL	R/W1S	0h	Set No-Reponse-Error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the No-Response-Error interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
26	SETFEINTLVL	R/W1S	0h	Set Framing-Error interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the Framing-Error interrupt level to the INT1 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
25	SETOEINTLVL	R/W1S	0h	Set Overrun-Error Interrupt Level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the Overrun-Error interrupt level to the INT1 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
24	SETPEINTLVL	R/W1S	0h	Set Parity Error interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the Parity error interrupt level to the INT1 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
23-19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17-16	RESERVED	R	0h	Reserved
15-14	RESERVED	R	0h	Reserved
13	SETIDINTLVL	R/W1S	0h	Set ID interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the ID interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
12-10	RESERVED	R	0h	Reserved
9	SETRXINTOVO	R/W1S	0h	Set Receiver interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the receiver interrupt level to the INT1 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
8	SETTXINTLVL	R/W1S	0h	Set Transmitter interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the transmitter interrupt level to the INT1 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
7	SETTOA3WUSINTLVL	R/W1S	0h	Set Timeout After 3 Wakeup Signals interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the timeout after 3 wakeup signals interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.

Table 40-20. SCISETINTLVL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	SETTOAWUSINTLVL	R/W1S	0h	Set Timeout After Wakeup Signal interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the the timeout after wakeup interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
5	RESERVED	R	0h	Reserved
4	SETTIMEOUTINTLVL	R/W1S	0h	Set Timeout interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the timeout interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
3-2	RESERVED	R	0h	Reserved
1	SETWAKEUPINTLVL	R/W1S	0h	Set Wake-up interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the Wake-up interrupt level to the INT1 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
0	SETBRKDTINTLVL	R/W1S	0h	Set Break-detect interrupt level. This bit is effective in SCI-compatible mode only. Writing to this bit maps the Break-detect interrupt level to the INT1 line. This field is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.

40.7.2.7 SCICLEARINTLVL Register (Offset = 18h) [Reset = 00000000h]

SCICLEARINTLVL is shown in Figure 40-33 and described in Table 40-21.

Return to the Summary Table.

The SCICLEARINTLVL register is used to map individual interrupt sources to the INT0 line.

Figure 40-33. SCICLEARINTLVL Register

31	30	29	28	27	26	25	24
CLRBEINTLVL	CLRPBEINTLVL	CLRCEINTLVL	CLRISFEINTLVL	CLRNREINTLVL	CLRFEINTLVL	CLROEINTLVL	CLRPEINTLVL
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	16
RESERVED					RESERVED	RESERVED	
R-0h					R-0h	R-0h	
15	14	13	12	11	10	9	8
RESERVED		CLRIDINTLVL	RESERVED			CLRRXINTLVL	CLRTXINTLVL
R-0h		R/W1C-0h	R-0h			R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
CLRTOA3WUSINTLVL	CLRTOAWUSINTLVL	RESERVED	CLRTIMEOUTINTLVL	RESERVED		CLRWAKEUPINTLVL	CLRBRKDTINTLVL
R/W1C-0h	R/W1C-0h	R-0h	R/W1C-0h	R-0h		R/W1C-0h	R/W1C-0h

Table 40-21. SCICLEARINTLVL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CLRBEINTLVL	R/W1C	0h	Clear Bit Error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the Bit Error interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
30	CLRPBEINTLVL	R/W1C	0h	Clear Physical Bus Error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the Physical Bus Error interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
29	CLRCEINTLVL	R/W1C	0h	Clear Checksum-error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the Checksum-error interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
28	CLRISFEINTLVL	R/W1C	0h	Clear Inconsistent-Sync-Field-Error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the Inconsistent-Sync-Field-Error interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.

Table 40-21. SCICLEARINTLVL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	CLRNREINTLVL	R/W1C	0h	Clear No-Reponse-Error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the No-Response-Error interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
26	CLRFEINTLVL	R/W1C	0h	Clear Framing-Error interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the Framing-Error interrupt level to the INT0 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
25	CLROEINTLVL	R/W1C	0h	Clear Overrun-Error Interrupt Level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the Overrun-Error interrupt level to the INT0 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
24	CLRPEINTLVL	R/W1C	0h	Clear Parity Error interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the Parity Error interrupt level to the INT0 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
23-19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17-16	RESERVED	R	0h	Reserved
15-14	RESERVED	R	0h	Reserved
13	CLRIDINTLVL	R/W1C	0h	Clear ID interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the ID interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
12-10	RESERVED	R	0h	Reserved
9	CLRRXINTLVL	R/W1C	0h	Clear Receiver interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the receiver interrupt level to the INT0 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
8	CLRTXINTLVL	R/W1C	0h	Clear Transmitter interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the transmitter interrupt level to the INT0 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.

Table 40-21. SCICLEARINTLVL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	CLRTOA3WUSINTLVL	R/W1C	0h	Clear Timeout After 3 Wakeup Signals interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the timeout after 3 wakeup signals interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
6	CLRTOAWUSINTLVL	R/W1C	0h	Clear Timeout After Wakeup Signal interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the the timeout after wakeup interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
5	RESERVED	R	0h	Reserved
4	CLRTIMEOUTINTLVL	R/W1C	0h	Clear Timeout interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the timeout interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
3-2	RESERVED	R	0h	Reserved
1	CLRWAKEUPINTLVL	R/W1C	0h	Clear Wake-up interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the Wake-up interrupt level to the INT0 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
0	CLRBRKDTINTLVL	R/W1C	0h	Clear Break-detect interrupt level. This bit is effective in SCI-compatible mode only. Writing to this bit maps the Break-detect interrupt level to the INT0 line. This field is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.

40.7.2.8 SCIFLR Register (Offset = 1Ch) [Reset = 00000904h]

SCIFLR is shown in Figure 40-34 and described in Table 40-22.

Return to the Summary Table.

The SCIFLR register indicates the current status of the various interrupt sources of the LIN module.

Figure 40-34. SCIFLR Register

31	30	29	28	27	26	25	24
BE	PBE	CE	ISFE	NRE	FE	OE	PE
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	IDRXFLAG	IDTXFLAG	RXWAKE	TXEMPTY	TXWAKE	RXRDY	TXRDY
R-0h	R/W1C-0h	R/W1C-0h	R-0h	R-1h	R/W-0h	R/W1C-0h	R-1h
7	6	5	4	3	2	1	0
TOA3WUS	TOAWUS	RESERVED	TIMEOUT	BUSY	IDLE	WAKEUP	BRKDT
R/W1C-0h	R/W1C-0h	R-0h	R/W1C-0h	R-0h	R-1h	R/W1C-0h	R/W1C-0h

Table 40-22. SCIFLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	BE	R/W1C	0h	Bit Error Flag. This bit is effective in LIN mode only. This bit is set when there has been a bit error. This is detected by the bit monitor in the internal bit monitor. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit - Reception of a new sync break This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No bit error detected. 1h (R/W) = Bit error detected.
30	PBE	R/W1C	0h	Physical Bus Error Flag. This bit is effective in LIN mode only. This bit is set when there has been a physical bus error. This is detected by the bit monitor in TED. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit - Reception of a new sync break Note: thie PBE will ony be flagged if no sync break can be generated. (because of a bus shortage to VBAT) or if no sync break delimeter can be generated (because of a bus shortage to GND). This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No physiscal bus error detected. 1h (R/W) = Physical bus error detected.

Table 40-22. SCIFLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
29	CE	R/W1C	0h	Checksum Error Flag. This bit is effective in LIN mode only. This bit is set when there is checksum error detected by a receiving node. The type of checksum to be used depends on the SCIGCR1.CTYPE bit. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit - Reception of a new sync break This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No Checksum error detected. 1h (R/W) = Checksum error detected.
28	ISFE	R/W1C	0h	Inconsistent Sync Field Error Flag. This bit is effective in LIN mode only. This bit is set when there has been an inconsistent Sync Field error detected by the synchronizer during header reception. See the 'Header Reception and Adaptive Baudrate' section for more information. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit - Reception of a new sync break This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No Inconsistent Sync Field error detected. 1h (R/W) = Inconsistent Sync Field error detected.
27	NRE	R/W1C	0h	No-Response Error Flag. This bit is effective in LIN mode only. This bit is set when there is no response to a Commander's header completed within TFRAME_MAX. This timeout period is applied for message frames of unknown length (identifiers 0 to 61). This error is detected by the synchronizer of the module. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit - Reception of a new sync break This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No No-Response error detected. 1h (R/W) = No-Response error detected.

Table 40-22. SCIFLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26	FE	R/W1C	0h	Framing error flag. This bit is effective in LIN or SCI-compatible mode. This bit is set when an expected stop bit is not found. In SCI compatible mode, only the first stop bit is checked. The missing stop bit indicates that synchronization with the start bit has been lost and that the character is incorrectly framed. Detection of a framing error causes the SCI to generate an error interrupt if the RXERR INT ENA bit is set. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit - Reception of a new character (SCI-compatible mode), or frame (LIN mode) In multibuffer mode the frame is defined in the SCIFORMAT register. Reset type: SYSRSn 0h (R/W) = No framing error detected. 1h (R/W) = Framing error detected.
25	OE	R/W1C	0h	Overrun error flag. This bit is effective in LIN or SCI-compatible mode. This bit is set when the transfer of data from SCIRXSHF to SCIRD overwrites unread data already in SCIRD or the RDy buffers. Detection of an overrun error causes the LIN to generate an error interrupt if the SET OE INT bit is one. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit Reset type: SYSRSn 0h (R/W) = No overrun error detected. 1h (R/W) = Overrun error detected.
24	PE	R/W1C	0h	Parity error flag. This bit is effective in LIN or SCI-compatible mode. This bit is set when a parity error is detected in the received data. In SCI address-bit mode, the parity is calculated on the data and address bit fields of the received frame. In idle-line mode, only the data is used to calculate parity. An error is generated when a character is received with a mismatch between the number of 1s and its parity bit. For more information on parity checking, see the 'SCI Global Control Register (SCIGCR1)' description. If the parity function is disabled (that is, SCIGCR1.2 = 0), the PE flag is disabled and read as 0. Detection of a parity error causes the LIN to generate an error interrupt if the SET PE INT bit = 1. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Reception of a new charcter (SCI-compatible mode) or frame (LIN mode) - Writing a 1 to this bit Reset type: SYSRSn 0h (R/W) = No parity error or parity disabled. 1h (R/W) = Parity error detected.
23-16	RESERVED	R	0h	Reserved
15	RESERVED	R	0h	Reserved

Table 40-22. SCIFLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	IDRXFLAG	R/W1C	0h	Identifier On Receive Flag. This bit is effective in LIN mode only. This flag is set once an identifier is received with an RX match and no ID-parity error. See the 'Message Filtering and Validation' section for more details. When this flag is set it indicates that a new valid identifier has been received on an RX match. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Reading the LINID register - Writing a 1 to this bit This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No valid ID received. 1h (R/W) = Valid ID RX received in LINID[23:16] on RX match.
13	IDTXFLAG	R/W1C	0h	Identifier On Transmit Flag. This bit is effective in LIN mode only. This flag is set once an identifier is received with a TX match and no ID-parity error. See the 'Message Filtering and Validation' section for more details. When this flag is set it indicates that a new valid identifier has been received on a TX match. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - RESET bit (SCIGCR0.0) - Setting SWnRESET - System reset - Reading the LINID register - Writing a 1 to this bit This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No valid ID received. 1h (R/W) = Valid ID received in LINID[23:16] on TX match.
12	RXWAKE	R	0h	Receiver wakeup detect flag. This bit is effective in SCI-compatible mode only. The SCI sets this bit to indicate that the data currently in SCIRD is an address. This bit is cleared by: - RESET bit - Setting the SWnRESET bit (SCIGCR1.7) - System reset - Receipt of a data frame This bit is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = The data in SCIRD is not an address. 1h (R/W) = The data in SCIRD is an address. See [1] Section 3.4.4, Sleep Mode for Multiprocessor Communication, on page 16 for more information on using the RXWAKE bit with sleep mode.

Table 40-22. SCIFLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	TXEMPTY	R	1h	Transmitter Empty flag. The value of this flag indicates the contents of the transmitter's buffer register(s) (SCITD/TDy) and shift register (SCITXSHF). In multibuffer mode, this flag indicates the value of the TDx registers and shift register (SCITXSHF). In non multibuffer mode, this flag indicates the value of LINTD0 (byte) and shift register (SCITXSHF). This bit is set by: - RESET bit (SCIGCR0.0) - Setting the SWnRESET bit (SCIGCR1.7) - System reset. Note: This bit does not cause an interrupt request. Reset type: SYSRSn 0h (R/W) = Compatible mode or LIN with no multibuffer: Transmitter buffer or shift register (or both) are loaded with data. In LIN mode using multibuffer mode: Multibuffer or shift register (or all) are loaded with data. 1h (R/W) = Compatible mode or LIN with no multibuffer: Transmitter buffer and shift registers are both empty. In LIN mode using multibuffer mode: Multibuffer and shift registers are all empty.
10	TXWAKE	R/W	0h	SCI transmitter wakeup method select. This bit is effective in SCI-compatible mode only. The TXWAKE bit controls whether the data in SCITD should be sent as an address or data frame using multiprocessor communication format. This bit is set to 1 or 0 by software before a byte is written to SCITD and is cleared by the SCI when data is transferred from SCITD to SCITXSHF or by a system reset. TXWAKE is not cleared by the SWnRESET bit (SCIGCR1.7). Reset type: SYSRSn 0h (R/W) = Address-bit mode: Frame to be transmitted will be data (address bit = 0). Idle-line mode: Frame to be transmitted will be data. 1h (R/W) = Address-bit mode: Frame to be transmitted will be an address (address bit=1). Idle-line mode: Following frame to be transmitted will be an address (writing a 1 to this bit followed by writing dummy data to the SCITD will result in a idle period of 11 bit periods before the next frame is transmitted).
9	RXRDY	R/W1C	0h	Receiver ready flag. In SCI compatibility mode, the receiver sets this bit to indicate that the SCIRD contains new data and is ready to be read by the CPU. In LIN mode, RXRDY is set once a valid frame is received in multibuffer mode, a valid frame being a message frame received with no errors. In non multibuffer mode RXRDY is set for each received byte and will be set for the last byte of the frame if there are no errors. The SCI/LIN generates a receive interrupt when RXRDY flag bit is set if the interrupt-enable bit is set (SCISETINT.9). RXRDY is cleared by: - RESET bit (SCIGCR0.0) - Setting the SWnRESET - System reset - Writing a 1 to this bit - Reading SCIRD in while in SCI compatibility mode - Reading last data byte RDy of the response in LIN mode Note: The RXRDY flag cannot be cleared by reading the corresponding interrupt offset in the SCIINTVECT0/1 register. Reset type: SYSRSn 0h (R/W) = No new data in SCIRD/RDy. 1h (R/W) = New data ready to be read from SCIRD.

Table 40-22. SCIFLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	TXRDY	R	1h	Transmitter buffer register ready flag. When set, this bit indicates that the transmit buffer(s) register (SCITD in compatibility mode and LINTD0, LINTD1 in MBUF mode) is/are ready to get another character from a CPU write. In SCI compatibility mode, writing data to SCITD automatically clears this bit. In LIN mode, this bit is cleared once byte 0 (TD0) is written to LINTD0. This bit is set after the data of the TX buffer are shifted into the SCITXSHF register. For devices with a DMA module, this event can trigger a transmit DMA event if the DMA enable bit is set. This bit is set to 1 by: - RESET bit (SCIGCR0.0) - Setting the SWnRESET (SCIGCR1.7) - System reset Note: The TXRDY flag cannot be cleared by reading the corresponding interrupt offset in the SCIINTVECT0/1 register. Note: The transmit interrupt request can be eliminated until the next series of data is written into the transmit buffers LINTD0 and LINTD1, by disabling the corresponding interrupt via the SCICLEARINT register or by disabling the transmitter via the TXENA bit (SCIGCR1.25=0). Reset type: SYSRSn 0h (R/W) = Compatible mode: SCITD is full. LIN mode: The multibuffers are full. 1h (R/W) = Compatible mode: SCITD is ready to receive the next character. LIN mode: The multibuffers are ready to receive the next character(s).
7	TOA3WUS	R/W1C	0h	Timeout After 3 Wakeup Signals flag. This bit is effective in LIN mode only. This flag is set if there is no Sync Break received after 3 wakeup signals and a period of 1.5 seconds have passed. Such expiration time is used before issuing another round of wakeup signals. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No timeout after 3 wakeup signals. 1h (R/W) = Timeout after 3 wakeup signals and 1.5s time.
6	TOAWUS	R/W1C	0h	Timeout After Wakeup Signal flag. This bit is effective in LIN mode only. This bit is set if there is no Sync Break received after a wakeup signal has been sent. A minimum of 150 ms expiration time is used before issuing another wakeup signal. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No timeout after one wakeup signal (150 ms). 1h (R/W) = Timeout after one wakeup signal.
5	RESERVED	R	0h	Reserved

Table 40-22. SCIFLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	TIMEOUT	R/W1C	0h	LIN Bus IDLE timeout flag. This bit is effective in LIN mode only. This bit is set if there is no LIN bus activity for at least 4 seconds. LIN bus activity being a transition from recessive to dominant. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No bus idle detected. 1h (R/W) = LIN bus idle detected.
3	BUSY	R	0h	Bus BUSY flag. This bit is effective in LIN mode and SCI-compatible mode. This bit indicates whether the receiver is in the process of receiving a frame. As soon as the receiver detects the beginning of a start bit, the BUSY bit is set to 1. When the reception of a frame is complete, the BUSY bit is cleared. If SET WAKEUP INT is set and power down is requested while this bit is set, the SCI/LIN automatically prevents low-power mode from being entered and generates wakeup interrupt. The BUSY bit is controlled directly by the SCI receiver but can be cleared by: - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset. Reset type: SYSRSn 0h (R/W) = Receiver is not currently receiving a frame. 1h (R/W) = Receiver is currently receiving a frame.
2	IDLE	R	1h	SCI receiver in idle state. This bit is effective in SCI-compatible mode only. While this bit is set, the SCI looks for an idle period to resynchronize itself with the bit stream. The receiver does not receive any data while the bit is set. The bus must be idle for 11 bit periods to clear this bit. The SCI enters this state: - After a system reset - Setting the SWnRESET bit (SCIGCR1.7) - After coming out of power down This bit is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = Idle period detected, the SCI is ready to receive. 1h (R/W) = Idle period not detected, the SCI will not receive any data.
1	WAKEUP	R/W1C	0h	Wake-up flag. This bit is effective in LIN mode only. This bit is set by the SCI/LIN when receiver or transmitter activity has taken the module out of power-down mode. An interrupt is generated if the SET WAKEUP INT bit (SCISETINT.1) is set. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register. - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Do not wake up from power-down mode. 1h (R/W) = Wake up from power-down mode.

Table 40-22. SCIFLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	BRKDT	R/W1C	0h	SCI break-detect flag. This bit is effective in SCI-compatible mode only. This bit is set when the SCI detects a break condition on the LINRX pin. A break condition occurs when the LINRX pin remains continuously low for at least 10 bits after a missing first stop bit, that is, after a framing error. Detection of a break condition causes the SCI to generate an error interrupt if the BRKDT INT ENA bit is set. The BRKDT bit is cleared by the following: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register. - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - By writing a 1 to this bit. This bit is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = No break condition detected. 1h (R/W) = Break condition detected.

40.7.2.9 SCIINTVECT0 Register (Offset = 20h) [Reset = 00000000h]

SCIINTVECT0 is shown in Figure 40-35 and described in Table 40-23.

Return to the Summary Table.

The SCIINTVECT0 register indicates the offset for the INT0 interrupt line.

Figure 40-35. SCIINTVECT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											INTVECT0				
R-0h											R-0h				

Table 40-23. SCIINTVECT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-5	RESERVED	R	0h	Reserved
4-0	INTVECT0	R	0h	Interrupt vector offset for INT0. This register indicates the offset for interrupt line INT0. A read to this register updates its value to the next highest priority pending interrupt in SCIFLR and clears the flag corresponding to the offset that was read. Note: The flags for the receive (SCIFLR.9) and the transmit (SCIFLR.8) interrupts cannot be cleared by reading the corresponding offset vector in this register (see detailed description in SCIFLR register). Reset type: SYSRSn

40.7.2.10 SCIINTVECT1 Register (Offset = 24h) [Reset = 00000000h]

SCIINTVECT1 is shown in Figure 40-36 and described in Table 40-24.

Return to the Summary Table.

The SCIINTVECT1 register indicates the offset for the INT1 interrupt line.

Figure 40-36. SCIINTVECT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											INTVECT1				
R-0h											R-0h				

Table 40-24. SCIINTVECT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-5	RESERVED	R	0h	Reserved
4-0	INTVECT1	R	0h	Interrupt vector offset for INT1. This register indicates the offset for interrupt line INT1. A read to this register updates its value to the next highest priority pending interrupt in SCIFLR and clears the flag corresponding to the offset that was read. Note: The flags for the receive (SCIFLR.9) and the transmit (SCIFLR.8) interrupts cannot be cleared by reading the corresponding offset vector in this register (see detailed description in SCIFLR register). Reset type: SYSRSn

40.7.2.11 SCIFORMAT Register (Offset = 28h) [Reset = 00000000h]

SCIFORMAT is shown in Figure 40-37 and described in Table 40-25.

Return to the Summary Table.

The SCIFORMAT register is used to set up the character and frame lengths.

Figure 40-37. SCIFORMAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED													LENGTH		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													CHAR		
R-0h													R/W-0h		

Table 40-25. SCIFORMAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18-16	LENGTH	R/W	0h	Frame length control bits. In LIN mode, these bits indicate the number of bytes in the response field from 1 to 8 bytes. In buffered SCI mode, these bits indicate the number of characters. When these bits are used to indicate LIN response length (SCIGCR1[0] = 1), then when there is an ID RX match, this value should be updated with the expected length of the response. In buffered SCI mode, these bits indicate the number of characters with SCIFORMAT[2:0] bits per character. i.e. these bits indicate the transmitter/receiver format for the number of characters: 1 to 8. There can be up to eight characters with eight bits each. Reset type: SYSRSn 0h (R/W) = The response field has 1 bytes/characters. 1h (R/W) = The response field has 2 bytes/characters. 2h (R/W) = The response field has 3 bytes/characters. 3h (R/W) = The response field has 4 bytes/characters. 4h (R/W) = The response field has 5 bytes/characters. 5h (R/W) = The response field has 6 bytes/characters. 6h (R/W) = The response field has 7 bytes/characters. 7h (R/W) = The response field has 8 bytes/characters.
15-3	RESERVED	R	0h	Reserved
2-0	CHAR	R/W	0h	Character length control bits. These bits are effective in SCI compatible mode only. These bits set the SCI character length from 1 to 8 bits. Note: In compatibility mode or buffered SCI mode, when data of fewer than eight bits in length is received, it is left justified in SCIRD/RDy and padded with trailing zeros. Data read from the SCIRD should be shifted by software to make the received data right justified. Note: Data written to the SCITD should be right justified but does not need to be padded with leading zeros. These bits are witable in SCI mode only. Reset type: SYSRSn 0h (R/W) = The character is 1 bits long. 1h (R/W) = The character is 2 bits long. 2h (R/W) = The character is 3 bits long. 3h (R/W) = The character is 4 bits long. 4h (R/W) = The character is 5 bits long. 5h (R/W) = The character is 6 bits long. 6h (R/W) = The character is 7 bits long. 7h (R/W) = The character is 8 bits long.

40.7.2.12 BRSR Register (Offset = 2Ch) [Reset = 00000000h]

BRSR is shown in Figure 40-38 and described in Table 40-26.

Return to the Summary Table.

The BRSR register is used to configure the baud rate of the LIN module.

Figure 40-38. BRSR Register

31	30	29	28	27	26	25	24
RESERVED	U			M			
R-0h	R/W-0h			R/W-0h			
23	22	21	20	19	18	17	16
SCI_LIN_PSH							
R/W-0h							
15	14	13	12	11	10	9	8
SCI_LIN_PSL							
R/W-0h							
7	6	5	4	3	2	1	0
SCI_LIN_PSL							
R/W-0h							

Table 40-26. BRSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30-28	U	R/W	0h	Superfractional Divider Selection. (U) These bits are an additional fractional part for the baudrate specification. These bits allow a super fine tuning of the fractional baudrate with 7 more intermediate values for each of the M fractional divider values. See the Superfractional Divider section for more details. Reset type: SYSRSn
27-24	M	R/W	0h	SCI/LIN 4-bit Fractional Divider Selection. (M) These bits are effective in LIN or SCI asynchronous mode. These bits are used to select a baud rate for the SCI/LIN module, and they are a fractional part for the baud rate specification. The M divider allows fine-tuning of the baud rate over the P prescaler with 15 additional intermediate values for each of the P integer values. Reset type: SYSRSn
23-16	SCI_LIN_PSH	R/W	0h	PRESCALER P (High Bits). SCI/LIN 24-bit Integer Prescaler Selection. These bits are used to select a baudrate for the SCI/LIN module. These bits are effective in LIN mode and SCI compatible mode. The SCI/LIN has an internally generated serial clock determined by the LIN module input clock and the prescalers P and M in this register. The SCI/LIN uses the 24-bit integer prescaler P value to select 1 of over 16,700,000 available baud rates. The additional 4-bit fractional prescaler M refines the baudate selection. Reset type: SYSRSn

Table 40-26. BRSR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	SCI_LIN_PSL	R/W	0h	PRESCALER P (Low Bits). SCI/LIN 24-bit Integer Prescaler Selection. These bits are used to select a baudrate for the SCI/LIN module. These bits are effective in LIN mode and SCI compatible mode. The SCI/LIN has an internally generated serial clock determined by the LIN module input clock and the prescalers P and M in this register. The SCI/LIN uses the 24-bit integer prescaler P value to select 1 of over 16,700,000 available baud rates. The additional 4-bit fractional prescaler M refines the baudate selection. Reset type: SYSRSn

40.7.2.13 SCIED Register (Offset = 30h) [Reset = 00000000h]

SCIED is shown in Figure 40-39 and described in Table 40-27.

Return to the Summary Table.

The SCIED register is a duplicate copy of SCIRD register that has no affect on the RXRDY flag for use with an emulator.

Figure 40-39. SCIED Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ED							
R-0h																								R-0h							

Table 40-27. SCIED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	ED	R	0h	Receiver Emulation Data. This bit is effective in SCI-compatible mode only. Reading SCIED(7-0) does not clear the RXRDY flag. This register should be used only by an emulator that must continually read the data buffer without affecting the RXRDY flag. Reset type: SYSRSn

40.7.2.14 SCIRD Register (Offset = 34h) [Reset = 00000000h]

SCIRD is shown in Figure 40-40 and described in Table 40-28.

Return to the Summary Table.

The SCIRD register is where received data is stored and can be read from.

Figure 40-40. SCIRD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RD							
R-0h																								R-0h							

Table 40-28. SCIRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	RD	R	0h	Received Data. This bit is effective in SCI-compatible mode only. When a frame has been completely received, the data in the frame is transferred from the receiver shift register SCIRXSHF to this register. As this transfer occurs, the RXRDY flag is set and a receive interrupt is generated if RX INT ENA (SCISETINT0.9) is set. When the data is read from SCIRD, the RXRDY flag is automatically cleared. When the SCI receives data that is fewer than eight bits in length, it loads the data into this register in a left justified format padded with trailing zeros. Therefore, your software should perform a logical shift on the data by the correct number of positions to make it right justified. Reset type: SYSRSn

40.7.2.15 SCITD Register (Offset = 38h) [Reset = 00000000h]

SCITD is shown in Figure 40-41 and described in Table 40-29.

Return to the Summary Table.

The SCITD register is where data to be transmitted is written to by application software.

Figure 40-41. SCITD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								TD							
R-0h																								R/W-0h							

Table 40-29. SCITD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	TD	R/W	0h	Transmit data This bit is effective in SCI-compatible mode only. Data to be transmitted is written to this register. The transfer of data from this register to the transmit shift register SCITXSHF sets the TXRDY flag (SCIFLR.8), which indicates that SCITD is ready to be loaded with another byte of data. Note: If TX INT ENA (SCISETINT.8) is set, this data transfer also causes an interrupt. Note: Data written to the SCIRD register that is fewer than eight bits long must be right justified, but it does not need to be padded with leading zeros. Reset type: SYSRSn

40.7.2.16 SCIPIO0 Register (Offset = 3Ch) [Reset = 00000000h]

SCIPIO0 is shown in Figure 40-42 and described in Table 40-30.

Return to the Summary Table.

The SCIPIO0 register is used to enable the LINTX and LINRX pins.

Figure 40-42. SCIPIO0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					TXFUNC	RXFUNC	RESERVED
R-0h					R/W-0h	R/W-0h	R-0h

Table 40-30. SCIPIO0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	TXFUNC	R/W	0h	Transmit pin function. This bit is effective in LIN or SCI mode. This bit defines the function of LINTX pin. Reset type: SYSRSn 0h (R/W) = LINTX pin is disabled. 1h (R/W) = LINTX pin is enabled.
1	RXFUNC	R/W	0h	Receive pin function. This bit is effective in LIN or SCI mode. This bit defines the function of the LINRX pin. Reset type: SYSRSn 0h (R/W) = LINRX pin is disabled. 1h (R/W) = LINRX pin is enabled.
0	RESERVED	R	0h	Reserved

40.7.2.17 SCIPIO2 Register (Offset = 44h) [Reset = 00000000h]

SCIPIO2 is shown in Figure 40-43 and described in Table 40-31.

Return to the Summary Table.

The SCIPIO2 register indicates the current status of the LINTX and LINRX pins.

Figure 40-43. SCIPIO2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					TXIN	RXIN	RESERVED
R-0h					R-0h	R-0h	R-0h

Table 40-31. SCIPIO2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	TXIN	R	0h	Transmit data in. This bit is effective in LIN or SCI-compatible mode. This bit contains the current value on the LINTX pin. Reset type: SYSRSn
1	RXIN	R	0h	Receive data in. This bit is effective in LIN or SCI-compatible mode. This bit contains the current value on the LINRX pin. Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

40.7.2.18 LINCOMP Register (Offset = 60h) [Reset = 00000000h]

LINCOMP is shown in Figure 40-44 and described in Table 40-32.

Return to the Summary Table.

The LINCOMPARE register is used to configure the sync delimeter and sync break extension.

Figure 40-44. LINCOMP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						SDEL		RESERVED					SBREAK		
R-0h						R/W-0h		R-0h					R/W-0h		

Table 40-32. LINCOMP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-10	RESERVED	R	0h	Reserved
9-8	SDEL	R/W	0h	2-bit Sync Delimiter compare. These bits are effective in LIN mode only. These bits are used to configure the number of Tbit for the sync delimiter in the sync field. The time delay calculation for the synchronization delimiter is: TSDEL = (SDEL + 1)Tbit These bits are writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = The sync delimiter has 1 Tbit. 1h (R/W) = The sync delimiter has 2 Tbit. 2h (R/W) = The sync delimiter has 3 Tbit. 3h (R/W) = The sync delimiter has 4 Tbit.
7-3	RESERVED	R	0h	Reserved
2-0	SBREAK	R/W	0h	3-bit Sync Break extend. LIN mode only. These bits are used to configure the number of Tbits for the sync break to extend the minimum 13 Tbit in the Sync Field to a maximum of 20 Tbit. The time delay calculation for the sync break is: TSYNBRK = 13Tbit + SBREAK x Tbit These bits are writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = The sync break has no additional Tbit. 1h (R/W) = The sync break has 1 additional Tbit. 2h (R/W) = The sync break has 2 additional Tbit. 3h (R/W) = The sync break has 3 additional Tbit. 4h (R/W) = The sync break has 4 additional Tbit. 5h (R/W) = The sync break has 5 additional Tbit. 6h (R/W) = The sync break has 6 additional Tbit. 7h (R/W) = The sync break has 7 additional Tbit.

40.7.2.19 LINRD0 Register (Offset = 64h) [Reset = 00000000h]

LINRD0 is shown in Figure 40-45 and described in Table 40-33.

Return to the Summary Table.

The LINRD0 register contains the lower 4 bytes of the received LIN frame data.

Figure 40-45. LINRD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RD0								RD1								RD2								RD3							
R-0h								R-0h								R-0h								R-0h							

Table 40-33. LINRD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RD0	R	0h	8-bit Receive Buffer 0 Each response data-byte that is received in the SCIRXSHFT register is transferred to the corresponding RDy register according to the number of bytes received. A read of this byte clears the RXDY byte. Note: RD<x-1> is equivalent to Data byte <x> of the LIN frame. Reset type: SYSRSn
23-16	RD1	R	0h	8-bit Receive Buffer 1. Each response data-byte that is received in the SCIRXSHFT register is transferred to the corresponding RDy register according to the number of bytes received. Reset type: SYSRSn
15-8	RD2	R	0h	8-bit Receive Buffer 2. Each response data-byte that is received in the SCIRXSHFT register is transferred to the corresponding RDy register according to the number of bytes received. Reset type: SYSRSn
7-0	RD3	R	0h	8-bit Receive Buffer 3. Each response data-byte that is received in the SCIRXSHFT register is transferred to the corresponding RDy register according to the number of bytes received. Reset type: SYSRSn

40.7.2.20 LINRD1 Register (Offset = 68h) [Reset = 00000000h]

LINRD1 is shown in Figure 40-46 and described in Table 40-34.

Return to the Summary Table.

The LINRD1 regsiter contains the upper 4 bytes of the received LIN frame data.

Figure 40-46. LINRD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RD4								RD5								RD6								RD7							
R-0h								R-0h								R-0h								R-0h							

Table 40-34. LINRD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RD4	R	0h	8-bit Receive Buffer 4 Each response data-byte that is received in the SCIRXSHFT register is transferred to the corresponding RDy register according to the number of bytes received. Reset type: SYSRSn
23-16	RD5	R	0h	8-bit Receive Buffer 5. Each response data-byte that is received in the SCIRXSHFT register is transferred to the corresponding RDy register according to the number of bytes received. Reset type: SYSRSn
15-8	RD6	R	0h	8-bit Receive Buffer 6. Each response data-byte that is received in the SCIRXSHFT register is transferred to the corresponding RDy register according to the number of bytes received. Reset type: SYSRSn
7-0	RD7	R	0h	8-bit Receive Buffer 7. Each response data-byte that is received in the SCIRXSHFT register is transferred to the corresponding RDy register according to the number of bytes received. Reset type: SYSRSn

40.7.2.21 LINMASK Register (Offset = 6Ch) [Reset = 00000000h]

LINMASK is shown in Figure 40-47 and described in Table 40-35.

Return to the Summary Table.

The LINMASK register is used to configure the masks used for filtering incoming ID messages for receive and transmit frames.

Figure 40-47. LINMASK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RXIDMASK								RESERVED								TXIDMASK							
R-0h								R/W-0h								R-0h								R/W-0h							

Table 40-35. LINMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	RXIDMASK	R/W	0h	Receive ID mask. This field is effective in LIN mode only.This 8-bit mask is used for filtering an incoming ID message and compare it to the ID-byte. A compare match of the received ID with the RX ID mask will set the ID RX flag and trigger and ID interrupt if enabled. A 0 bit in the mask indicates that bit is compared to the ID-byte. A 1 bit in the mask indicates that that bit is filtered and therefore not used in the compare. When HGENCTRL is set to 1, this field must be set to 0xFF if the complete ID must be compared. Reset type: SYSRSn
15-8	RESERVED	R	0h	Reserved
7-0	TXIDMASK	R/W	0h	Transmit ID mask. This field is effective in LIN mode only. This 8-bit mask is used for filtering an incoming ID message and comparing it to the ID-byte. A compare match of the received ID with the TX ID Mask will set the ID TX flag and trigger an ID interrupt if enabled. A 0 bit in the mask indicates that bit is compared to the ID-byte. A 1 bit in the mask indicates that bit is filtered and therefore not used for the compare. When HGENCTRL is set to 1, this field must be set to 0xFF if the complete ID must be compared. Reset type: SYSRSn

40.7.2.22 LINID Register (Offset = 70h) [Reset = 00000000h]

LINID is shown in Figure 40-48 and described in Table 40-36.

Return to the Summary Table.

The LINID register contains the identification fields for LIN communication.
NOTE: For software compatibility with future LIN modules, the HGEN CTRL bit must be set to 1, the RX ID MASK field must be set to FFh, and the TX ID MASK field must be set to FFh.

Figure 40-48. LINID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								RECEIVEDID							
R-0h								R-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDRESPONDERTASKBYTE								IDBYTE							
R/W-0h								R/W-0h							

Table 40-36. LINID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	RECEIVEDID	R	0h	Received ID. This bit is effective in LIN mode only. This byte contains the current message identifier. During header reception the received ID is copied from the SCIRXSHF register to this byte if there is no ID-parity error and there has been an RX/TX match. Note: If a framing error (FE) is detected during ID reception, the received ID will also not be copied to the LINID register. Reset type: SYSRSn
15-8	IDRESPONDERTASKBYTE	R/W	0h	ID Responder Task byte. This field is effective in LIN mode only. This byte contains the identifier to which the received ID of an incoming header will be compared in order to decide whether a RX response, a TX response, or no action needs to be done by the LIN node. These bits are writable in LIN mode only. Reset type: SYSRSn
7-0	IDBYTE	R/W	0h	ID byte. This field is effective in LIN mode only. This byte is the LIN mode message ID. On a Commander node, a write to this register by the CPU initiates a header transmission. For a Responder task, this byte is used for message filtering when HGENCTRL (SCIGCR1.12) is '0'. These bits are writable in LIN mode only. Reset type: SYSRSn

40.7.2.23 LINTD0 Register (Offset = 74h) [Reset = 00000000h]

LINTD0 is shown in Figure 40-49 and described in Table 40-37.

Return to the Summary Table.

The LINTD0 register contains the lower 4 bytes of the data to be transmitted.
NOTE: TD<x-1> is equivalent to Data byte <x> of the LIN frame.

Figure 40-49. LINTD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TD0								TD1								TD2								TD3							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 40-37. LINTD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TD0	R/W	0h	8-bit Transmit Buffer 0. Byte 0 to be transmitted is written into this register and then copied to SCITXSHF for transmission. Once byte 0 is written in TDO buffer, transmission will be initiated. Reset type: SYSRSn
23-16	TD1	R/W	0h	8-bit Transmit Buffer 1. Byte 1 to be transmitted is written into this register and then copied to SCITXSHF for transmission. Reset type: SYSRSn
15-8	TD2	R/W	0h	8-bit Transmit Buffer 2. Byte 2 to be transmitted is written into this register and then copied to SCITXSHF for transmission. Reset type: SYSRSn
7-0	TD3	R/W	0h	8-bit Transmit Buffer 3. Byte 3 to be transmitted is written into this register and then copied to SCITXSHF for transmission. Reset type: SYSRSn

40.7.2.24 LINTD1 Register (Offset = 78h) [Reset = 00000000h]

LINTD1 is shown in Figure 40-50 and described in Table 40-38.

Return to the Summary Table.

The LINTD1 register contains the upper 4 bytes of the data to be transmitted.
NOTE: TD<x-1> is equivalent to Data byte <x> of the LIN frame.

Figure 40-50. LINTD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TD4								TD5								TD6								TD7							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 40-38. LINTD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TD4	R/W	0h	8-bit Transmit Buffer 4. Byte4 to be transmitted is written into this register and then copied to SCITXSHF for transmission. Reset type: SYSRSn
23-16	TD5	R/W	0h	8-bit Transmit Buffer 5. Byte 5 to be transmitted is written into this register and then copied to SCITXSHF for transmission. Reset type: SYSRSn
15-8	TD6	R/W	0h	8-bit Transmit Buffer 6. Byte 6 to be transmitted is written into this register and then copied to SCITXSHF for transmission. Reset type: SYSRSn
7-0	TD7	R/W	0h	8-bit Transmit Buffer 7. Byte 7 to be transmitted is written into this register and then copied to SCITXSHF for transmission. Reset type: SYSRSn

40.7.2.25 MBRSR Register (Offset = 7Ch) [Reset = 00000DACh]

MBRSR is shown in Figure 40-51 and described in Table 40-39.

Return to the Summary Table.

The MBRSR register is used to configure the expected maximum baud rate of the LIN network.

Figure 40-51. MBRSR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			MBR												
R-0h																			R/W-DACh											

Table 40-39. MBRSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	Reserved
12-0	MBR	R/W	DACh	Maximum Baud Rate Prescaler. This field is effective in LIN mode only. This 13-bit prescaler is used during the synchronization phase (see the 'Header Reception and Adaptive Baudrate' section) of a responder module if the ADAPT bit is set. In this way, a SCI/LIN responder using automatic or select bit rate modes detects any LIN bus legal rate automatically if the measured baud rate is within ±10% of the programmed baud rate. The MBR value should be programmed to allow a maximum baud rate that is not more than 10% above the expected operating baud rate in the LIN network. Otherwise a s 0x00 data byte could mistakenly be detected as sync break. The default value is for a 70MHz VCLK and ~18kbps expected baud rate. (0xDAC). This MBR prescaler is used by the wake-up and idle time counters for a constant expiration time relative to a 20kHz rate. MBR = VCLK Frequency / (1.1*Expected Baud Rate Frequency) Note: The MBR field must be written with a 13-bit value. If the calculated MBR exceeds 2^{13} = 8192, either the baud rate or the VCLK frequency must be adjusted for valid operation. Reset type: SYSRSn

40.7.2.26 IODFTCTRL Register (Offset = 90h) [Reset = 00000500h]

IODFTCTRL is shown in Figure 40-52 and described in Table 40-40.

Return to the Summary Table.

The IODFTCTRL register is used to emulate various error and test conditions.

Figure 40-52. IODFTCTRL Register

31	30	29	28	27	26	25	24
BERRENA	PBERRENA	CERRENA	ISFERRENA	RESERVED	FERRENA	PERRENA	BRKDTERRENA
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED			PINSAMPLEMASK		TXSHIFT		
R/W-0h			R/W-0h		R/W-0h		
15	14	13	12	11	10	9	8
RESERVED				IODFTENA			
R-0h				R/W-5h			
7	6	5	4	3	2	1	0
RESERVED						LPBENA	RXPENA
R-0h						R/W-0h	R/W-0h

Table 40-40. IODFTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	BERRENA	R/W	0h	Bit Errror Enable bit. This bit is effective in LIN mode only. This bit is used to create a Bit error. When this bit is set, the bit received is ORed with 1 and passed to the Bit monitor circuitry. Reset type: SYSRSn
30	PBERRENA	R/W	0h	Physical Bus Error Enable bit. This bit is effective in LIN mode only. This bit is used to create a Physical Bus Error. When this bit is set, the bit received during Sync Break field transmission is ORed with 1 and passed to the Bit monitor circuitry Reset type: SYSRSn
29	CERRENA	R/W	0h	Checksum Error Enable bit. This bit is effective in LIN mode only. This bit is used to create a checksum error. When this bit is set, the polarity of the CTYPE (checksum type) in the receive checksum calculator is changed so that a checksum error is generated. Reset type: SYSRSn
28	ISFERRENA	R/W	0h	Inconsistent Sync Field Error Enable bit. This bit is effective in LIN mode only. This bit is used to create an ISF error. When this bit is set, the bit widths in the sync field are varied so that the ISF check fails and the error flag is set. Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26	FERRENA	R/W	0h	This bit is used to create a Frame Error. This bit is effective in SCI-compatible mode only. When this bit is set, the stop bit received is ANDed with '0' and passed to the stop bit check circuitry. Reset type: SYSRSn

Table 40-40. IODFTCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	PERRENA	R/W	0h	Compatible Mode only This bit is effective in SCI-compatible mode only. This bit is used to create a Parity Error. When this bit is set, in compatible mode, the parity bit received is toggled so that a parity error occurs. Reset type: SYSRSn
24	BRKDTERRENA	R/W	0h	Compatible Mode only This bit is effective in SCI-compatible mode only. This bit is used to create BRKDT error (SCI mode only). When this bit is set, the stop bit of the frame is ANDed with '0' and passed to the RSM so that a frame error occurs. Then the RX Pin is forced to continuous low for 10 Tbits so that a BRKDT error occurs. Reset type: SYSRSn
23-21	RESERVED	R/W	0h	Reserved
20-19	PINSAMPLEMASK	R/W	0h	Pin sample mask. These bits define the sample number at which the TX Pin value that is being transmitted will be inverted to verify the receive pin samples correctly with the majority detection circuitry. Note: During IODFT mode testing for the pin sample mask, the prescalar P must be programmed to be greater than 2. Reset type: SYSRSn 0h (R/W) = No Mask 1h (R/W) = Invert the TX Pin value at TBIT_CENTER 2h (R/W) = Invert the TX Pin value at TBIT_CENTER + SCLK 3h (R/W) = Invert the TX Pin value at TBIT_CENTER + 2 SCLK
18-16	TXSHIFT	R/W	0h	Transmit shift. These bits define the delay by which the value on LINTX is delayed so that the value on LINRX is asynchronous. (Not applicable to Start Bit) Reset type: SYSRSn 0h (R/W) = No Delay 1h (R/W) = Delay by 1 SCLK 2h (R/W) = Delay by 2 SCLK 3h (R/W) = Delay by 3 SCLK 4h (R/W) = Delay by 4 SCLK 5h (R/W) = Delay by 5 SCLK 6h (R/W) = Delay by 6 SCLK 7h (R/W) = Delay by 7 SCLK
15-12	RESERVED	R	0h	Reserved
11-8	IODFTENA	R/W	5h	IO DFT Enable Key This field is used to enable the IODFT mode of the SCI/LIN module for testing. Reset type: SYSRSn 0h (R/W) = IODFT is disabled 1h (R/W) = IODFT is disabled 2h (R/W) = IODFT is disabled 3h (R/W) = IODFT is disabled 4h (R/W) = IODFT is disabled 5h (R/W) = IODFT is disabled 6h (R/W) = IODFT is disabled 7h (R/W) = IODFT is disabled 8h (R/W) = IODFT is disabled 9h (R/W) = IODFT is disabled Ah (R/W) = IODFT is enabled Bh (R/W) = IODFT is disabled Ch (R/W) = IODFT is disabled Dh (R/W) = IODFT is disabled Eh (R/W) = IODFT is disabled Fh (R/W) = IODFT is disabled
7-2	RESERVED	R	0h	Reserved

Table 40-40. IODFTCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	LPBENA	R/W	0h	Module loopback enable. In analog loopback mode the complete communication path through the I/Os can be tested, whereas in digital loopback mode the I/O buffers are excluded from this path. Reset type: SYSRSn 0h (R/W) = Digital loopback is enabled. 1h (R/W) = Analog loopback is enabled in module I/O DFT mode (when IODFTENA = 1010)
0	RXPENA	R/W	0h	Module Analog loopback through receive pin enable. This bit defines whether the I/O buffers for the transmit or the receive pin are included in the communication path in analog loopback mode only. Reset type: SYSRSn 0h (R/W) = Analog loopback through the transmit pin is enabled. 1h (R/W) = Analog loopback through the receive pin is enabled.

40.7.2.27 LIN_GLB_INT_EN Register (Offset = E0h) [Reset = 00000000h]

LIN_GLB_INT_EN is shown in Figure 40-53 and described in Table 40-41.

Return to the Summary Table.

The LIN_GLB_INT_EN register is used to enable the INT0 and INT1 interrupt lines to propagate to the PIE block.

Figure 40-53. LIN_GLB_INT_EN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						GLBINT1_EN	GLBINT0_EN
R-0h						R/W-0h	R/W-0h

Table 40-41. LIN_GLB_INT_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	GLBINT1_EN	R/W	0h	Global Interrupt Enable for LIN INT1. This bit determines whether the INT1 interrupt line generates an interrupt to the PIE or not Reset type: SYSRSn 0h (R/W) = LIN INT1 line does not generate an interrupt to the PIE. 1h (R/W) = LIN INT1 line generates an interrupt to the PIE if an enabled interrupt condition occurs.
0	GLBINT0_EN	R/W	0h	Global Interrupt Enable for LIN INT0. This bit determines whether the INT0 interrupt line generates an interrupt to the PIE or not. Reset type: SYSRSn 0h (R/W) = LIN INT0 line does not generate an interrupt to the PIE. 1h (R/W) = LIN INT0 line generates an interrupt to the PIE if an enabled interrupt condition occurs.

40.7.2.28 LIN_GLB_INT_FLG Register (Offset = E4h) [Reset = 00000000h]

LIN_GLB_INT_FLG is shown in Figure 40-54 and described in Table 40-42.

Return to the Summary Table.

The LIN_GLB_INT_FLG register contains the current status of the INT0 and INT1 flags.

Figure 40-54. LIN_GLB_INT_FLG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						INT1_FLG	INT0_FLG
R-0h						R-0h	R-0h

Table 40-42. LIN_GLB_INT_FLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	INT1_FLG	R	0h	Global Interrupt Flag for LIN INT1. This bit indicates if an interrupt was generated to the PIE due to an enabled interrupt on the INT1 interrupt line. Refer to the LIN Interrupt Status Register for the condition that generated the interrupt. This bit can be cleared by writing a 1 to the corresponding bit in the LIN_GLB_INT_CLR register. Reset type: SYSRSn 0h (R/W) = No interrupt is active on the INT1 line. 1h (R/W) = An interrupt was generated due to an enabled interrupt on the INT1 interrupt line.
0	INT0_FLG	R	0h	Global Interrupt Flag for LIN INT0. This bit indicates if an interrupt was generated to the PIE due to an enabled interrupt on the INT0 interrupt line. Refer to the LIN Interrupt Status Register for the condition that generated the interrupt. This bit can be cleared by writing a 1 to the corresponding bit in the LIN_GLB_INT_CLR register. Reset type: SYSRSn 0h (R/W) = No interrupt is active on the INT0 line. 1h (R/W) = An interrupt was generated due to an enabled interrupt on the INT0 interrupt line.

40.7.2.29 LIN_GLB_INT_CLR Register (Offset = E8h) [Reset = 00000000h]

LIN_GLB_INT_CLR is shown in Figure 40-55 and described in Table 40-43.

Return to the Summary Table.

The LIN_GLB_INT_CLR register is used to clear the interrupt flags in LIN_GLB_INT_FLG register.

Figure 40-55. LIN_GLB_INT_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						INT1_FLG_CLR	INT0_FLG_CLR
R-0h						R/W1C-0h	R/W1C-0h

Table 40-43. LIN_GLB_INT_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	INT1_FLG_CLR	R/W1C	0h	Global Interrupt flag clear for LIN INT1. This bit is used to clear the corresponding bit in the LIN_GLB_INT_FLG register. Write 1 to clear the INT1_FLG bit. Writing 0 has no effect. Reset type: SYSRSn
0	INT0_FLG_CLR	R/W1C	0h	Global Interrupt flag clear for LIN INT0. This bit is used to clear the corresponding bit in the LIN_GLB_INT_FLG register. Write 1 to clear the INT0_FLG bit. Writing 0 has no effect. Reset type: SYSRSn

Chapter 41

Serial Peripheral Interface (SPI)

This chapter describes the serial peripheral interface (SPI) which is a high-speed synchronous serial input and output (I/O) port that allows a serial bit stream of programmed length (one to 16 bits) to be shifted into and out of the device at a programmed bit-transfer rate. The SPI is normally used for communications between the MCU controller and external peripherals or another controller. Typical applications include external I/O or peripheral expansion using devices such as shift registers, display drivers, and analog-to-digital converters (ADCs). Multi-device communications are supported by the controller or peripheral operation of the SPI. The port supports a 16-level, receive and transmit FIFO for reducing CPU servicing overhead.

41.1 Introduction

41.1.1 Features

The SPI module features include:

- SPIPOCI: SPI peripheral-output/controller-input pin
- SPIPICO: SPI peripheral-input/controller-output pin
- $\overline{\text{SPIPTE}}$: SPI peripheral transmit-enable pin
- SPICLK: SPI serial-clock pin

Note

All four pins can be used as GPIO if the SPI module is not used.

- Two operational modes: Controller and Peripheral
- Baud rate: 125 different programmable rates. The maximum baud rate that can be employed is limited by the maximum speed of the I/O buffers used on the SPI pins. See the device data sheet for more details.
- Data word length: 1 to 16 data bits
- Four clocking schemes (controlled by clock polarity and clock phase bits) include:
  - Falling edge without phase delay: SPICLK active-high. SPI transmits data on the falling edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal.
  - Falling edge with phase delay: SPICLK active-high. SPI transmits data one half-cycle ahead of the falling edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
  - Rising edge without phase delay: SPICLK inactive-low. SPI transmits data on the rising edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
  - Rising edge with phase delay: SPICLK inactive-low. SPI transmits data one half-cycle ahead of the rising edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal.
- Simultaneous receive and transmit operation (transmit function can be disabled in software)
- Transmitter and receiver operations are accomplished through either interrupt- driven or polled algorithm
- Delayed transmit control
- 16-level transmit/receive FIFO
- RTDMA support
- High-speed mode
- 3-wire SPI mode
- $\overline{\text{SPIPTE}}$ inversion for digital audio interface receive mode on devices with two SPI modules

41.1.2 Block Diagram

Figure 41-1 shows the SPI CPU interfaces.

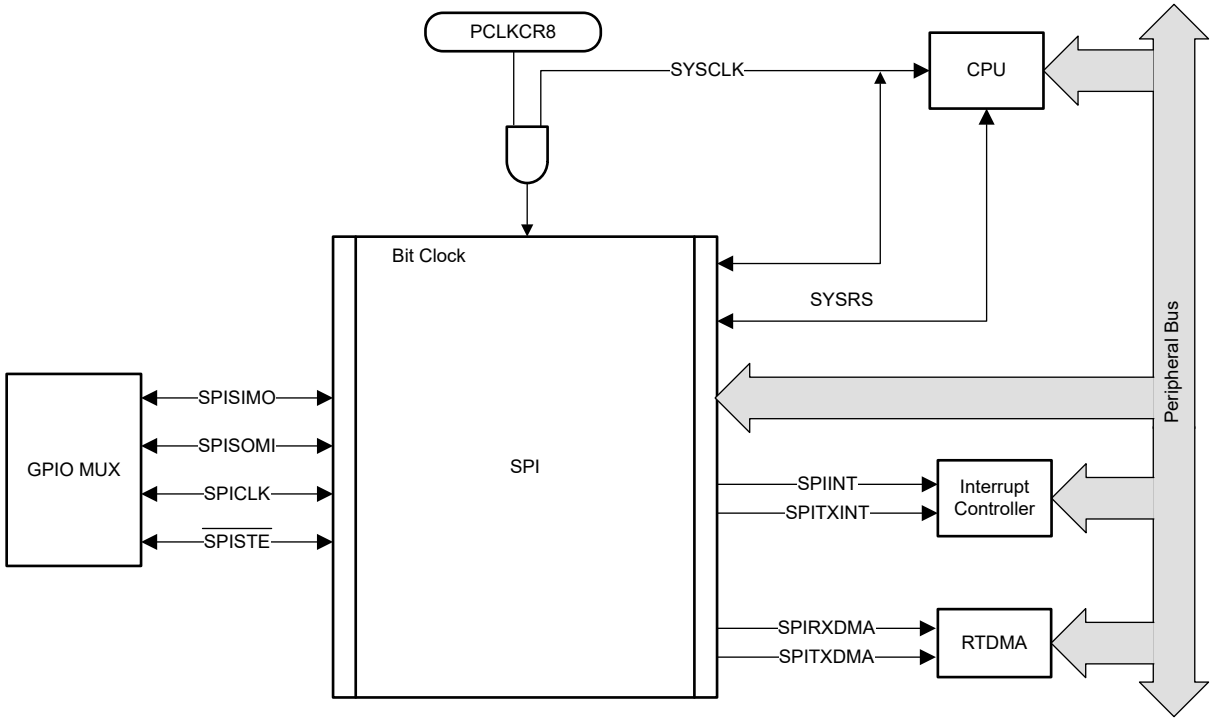


Figure 41-1. SPI CPU Interface

41.2 System-Level Integration

This section describes the various functionality that is applicable to the device integration. These features require configuration of other modules in the device that are not within the scope of this chapter.

41.2.1 SPI Module Signals

Table 41-1 classifies and provides a summary of the SPI module signals.

Table 41-1. SPI Module Signal Summary

Signal Name	Description
External Signals	
SPICLK	SPI clock
SPIPICO	SPI peripheral in, controller out
SPIPOCI	SPI peripheral out, controller in
$\overline{\text{SPIPTE}}$	SPI peripheral transmit enable
Control	
SPI Clock Rate	SYSCLK
Interrupt Signals	
SPIINT/SPIRXINT	Transmit interrupt/ Receive Interrupt in non FIFO mode (referred to as SPIINT)
	Receive interrupt in FIFO mode
SPITXINT	Transmit interrupt in FIFO mode
RTDMA Triggers	
SPITXDMA	Transmit request to RTDMA
SPIRXDMA	Receive request to RTDMA

Special Considerations

The $\overline{\text{SPIPTE}}$ signal provides the ability to gate any spurious clock and data pulses when the SPI is in peripheral mode. A HIGH logic signal on $\overline{\text{SPIPTE}}$ does not allow the peripheral to receive data. This prevents the SPI peripheral from losing synchronization with the controller. TI does not recommend that the $\overline{\text{SPIPTE}}$ always be tied to the active state.

If the SPI peripheral does ever lose synchronization with the controller, toggling SPISWRESET resets the internal bit counter as well as the various status flags in the module. By resetting the bit counter, the SPI interprets the next clock transition as the first bit of a new transmission. The register bit fields that are reset by SPISWRESET are found in .

Configuring a GPIO to Emulate $\overline{\text{SPIPTE}}$

In many systems, a SPI controller can be connected to multiple SPI peripherals using multiple instances of $\overline{\text{SPIPTE}}$. Though this SPI module does not natively support multiple $\overline{\text{SPIPTE}}$ signals, it is possible to emulate this behavior in software using GPIOs. In this configuration, the SPI must be configured as the controller. Rather than using the GPIO Mux to select $\overline{\text{SPIPTE}}$, the application can configure pins to be GPIO outputs, one GPIO per SPI peripheral. Before transmitting any data, the application can drive the desired GPIO to the active state. Immediately after the transmission has been completed, the GPIO chip select can be driven to the inactive state. This process can be repeated for many peripherals that share the SPICLK, SPIPICO, and SPIPOCI lines.

41.2.2 Configuring Device Pins

The GPIO mux registers must be configured to connect this peripheral to the device pins. To avoid glitches on the pins, the GPyGMUX bits must be configured first (while keeping the corresponding GPyMUX bits at the default of zero), followed by writing the GPyMUX register to the desired value.

Some IO functionality is defined by GPIO register settings independent of this peripheral. For input signals, the GPIO input qualification must be set to asynchronous mode by setting the appropriate GPxQSELn register bits to 11b. The internal pullups can be configured in the GPyPUD register.

See the *General-Purpose Input/Output (GPIO)* chapter for more details on GPIO mux and settings.

41.2.2.1 GPIOs Required for High-Speed Mode

To enable the high-speed enhancements, set SPICCR.HS_MODE to 1. Make sure that the capacitive loading on the pin does not exceed the value stated in the device data sheet.

When not operating in high-speed mode or if the capacitive loading on the pins exceed the value stated in the device data sheet, SPICCR.HS_MODE can be set to 0.

41.2.3 SPI Interrupts

This section includes information on the available interrupts present in the SPI module. The SPI module contains two interrupt lines: SPIINT/SPIRXINT and SPITXINT. When the SPI is operating in non-FIFO mode, all available interrupts are routed together to generate the single SPIINT interrupt. When FIFO mode is used, both SPIRXINT and SPITXINT can be generated.

SPIINT/SPIRXINT

When the SPI is operating in non-FIFO mode, the interrupt generated is called SPIINT. If FIFO enhancements are enabled, the interrupt is called SPIRXINT. These interrupts share the same interrupt vector in the Interrupt Controller block.

In non-FIFO mode, two conditions can trigger an interrupt: a transmission is complete (INT_FLAG), or there is overrun in the receiver (OVERRUN_FLAG). Both of these conditions share the same interrupt vector: SPIINT.

The transmission complete flag (INT_FLAG) indicates that the SPI has completed sending or receiving the last bit and is ready to be serviced. At the same time this bit is set, the received character is placed in the receiver buffer (SPIRXBUF). The INT_FLAG generates an interrupt on the SPIINT vector if the SPIINTENA bit is set.

The receiver overrun flag (OVERRUN_FLAG) indicates that a transmit or receive operation has completed before the previous character has been read from the buffer. The OVERRUN_FLAG generates an interrupt on the SPIINT vector if the OVERRUNINTENA bit is set and OVERRUN_FLAG was previously cleared.

In FIFO mode, the SPI can interrupt the CPU upon a match condition between the current receive FIFO status (RXFFST) and the receive FIFO interrupt level (RXFFIL). If RXFFST is greater than or equal to RXFFIL, the receive FIFO interrupt flag (RXFFINT) is set. SPIRXINT is triggered in the Interrupt Controller block, if RXFFINT is set and the receive FIFO interrupt is enabled (RXFFIENA = 1).

SPITXINT

The SPITXINT interrupt is not available when the SPI is operating in non-FIFO mode.

In FIFO mode, the SPITXINT behavior is similar to the SPIRXINT. SPITXINT is generated upon a match condition between the current transmit FIFO status (TXFFST) and the transmit FIFO interrupt level (TXFFIL). If TXFFST is less than or equal to TXFFIL, the transmit FIFO interrupt flag (TXFFINT) is set. SPITXINT is triggered in the Interrupt Controller block, if TXFFINT is set and the transmit FIFO interrupt is enabled in the SPI module (TXFFIENA = 1).

Figure 41-2 and Table 41-2 show how these control bits influence the SPI interrupt generation.

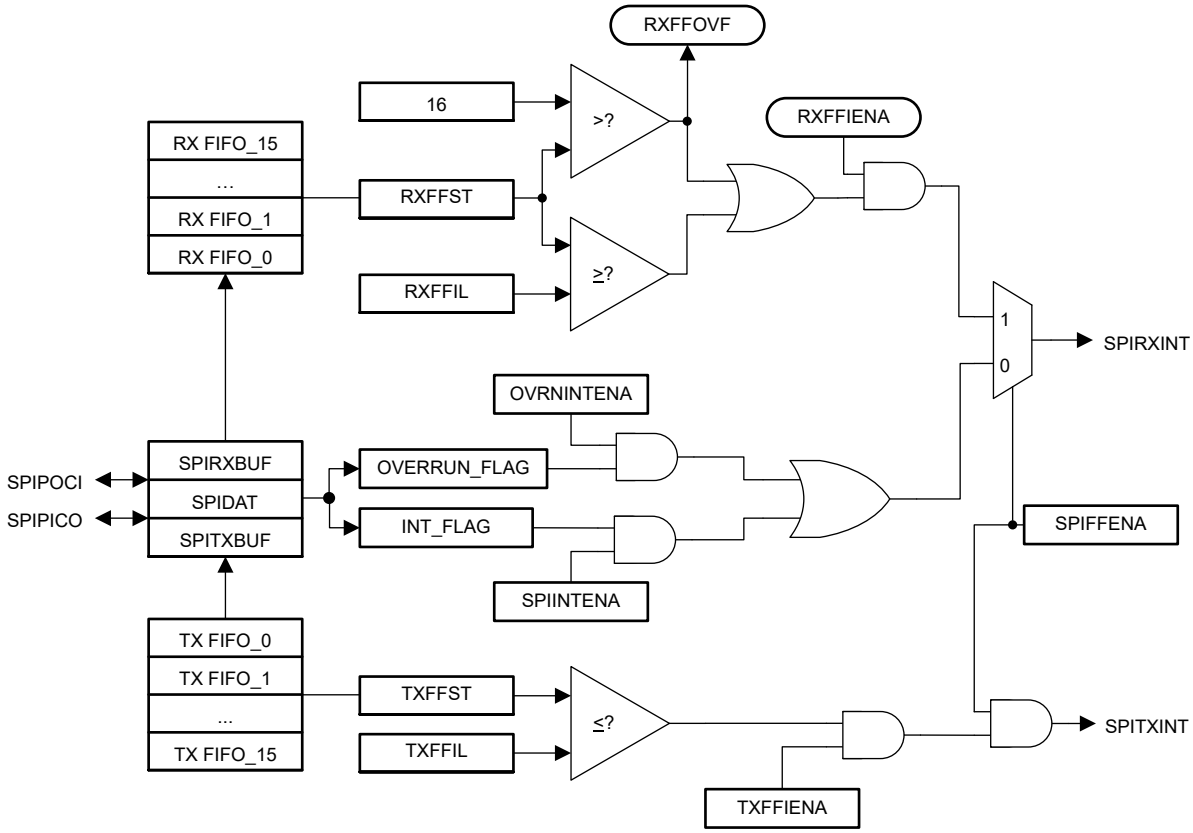


Figure 41-2. SPI Interrupt Flags and Enable Logic Generation

Table 41-2. SPI Interrupt Flag Modes

FIFO Options	SPI Interrupt Source	Interrupt Flags	Interrupt Enables	FIFO Enable (SPIFFENA)	Interrupt Line (1)
SPI without FIFO	Receive overrun	RXOVRN	OVRNINTENA	0	SPIRXINT
	Data receive	SPIINT	SPIINTENA	0	SPIRXINT
	Transmit empty	SPIINT	SPIINTENA	0	SPIRXINT
SPI FIFO mode	FIFO receive	RXFFIL	RXFFIENA	1	SPIRXINT
	Transmit empty	TXFFIL	TXFFIENA	1	SPITXINT

(1) In non-FIFO mode, SPIRXINT is the same name as the SPIINT interrupt in C29x devices.

41.2.4 RTDMA Support

Both the CPU and RTDMA have access to the SPI data registers using the internal peripheral bus. This access is limited to 16-bit register reads and writes. Each SPI module can generate two RTDMA events, SPITXDMA and SPIRXDMA. The RTDMA events are controlled by configuring the SPIFFTX.TXFFIL and SPIFFRX.RXFFIL appropriately. SPITXDMA activates when TXFFST is less than the interrupt level (TXFFIL). SPIRXDMA activates when RXFFST is greater than or equal to the interrupt level (RXFFIL).

The SPI must have FIFO enhancements enabled for the RTDMA triggers to be generated.

For more information on configuring the SPI for RTDMA transfers, refer to Section 41.3.8.

Figure 41-3 is a block diagram showing the RTDMA trigger generation from the SPI module.

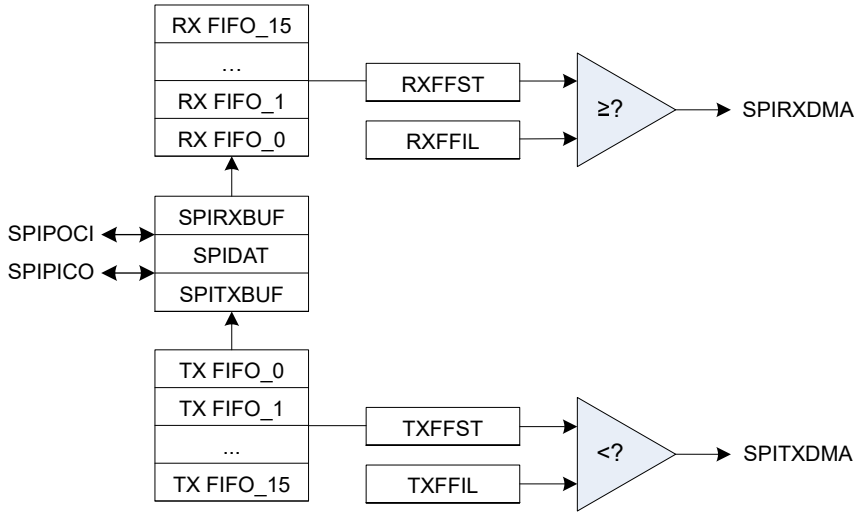


Figure 41-3. SPI RTDMA Trigger Diagram

41.3 SPI Operation

This section describes the various modes of operation of the SPI. Included are explanations of the operational modes, interrupts, data format, clock sources, and initialization. Typical timing diagrams for data transfers are given.

41.3.1 Introduction to Operation

Figure 41-4 shows typical connections of the SPI for communications between two controllers: a controller and a peripheral.

The controller transfers data by sending the SPICLK signal. For both the peripheral and the controller, data is shifted out of the shift registers on one edge of the SPICLK and latched into the shift register on the opposite SPICLK clock edge. If the CLK_PHASE bit is high, data is transmitted and received a half-cycle before the SPICLK transition. As a result, both controllers send and receive data simultaneously. The application software determines whether the data is meaningful or dummy data. There are three possible methods for data transmission:

- Controller sends data; peripheral sends dummy data.
- Controller sends data; peripheral sends data.
- Controller sends dummy data; peripheral sends data.

The controller can initiate data transfer at any time because the controller controls the SPICLK signal. The software, however, determines how the controller detects when the peripheral is ready to broadcast data.

The SPI operates in controller or peripheral mode. The CONTROLLER_PERIPHERAL bit selects the operating mode and the source of the SPICLK signal.

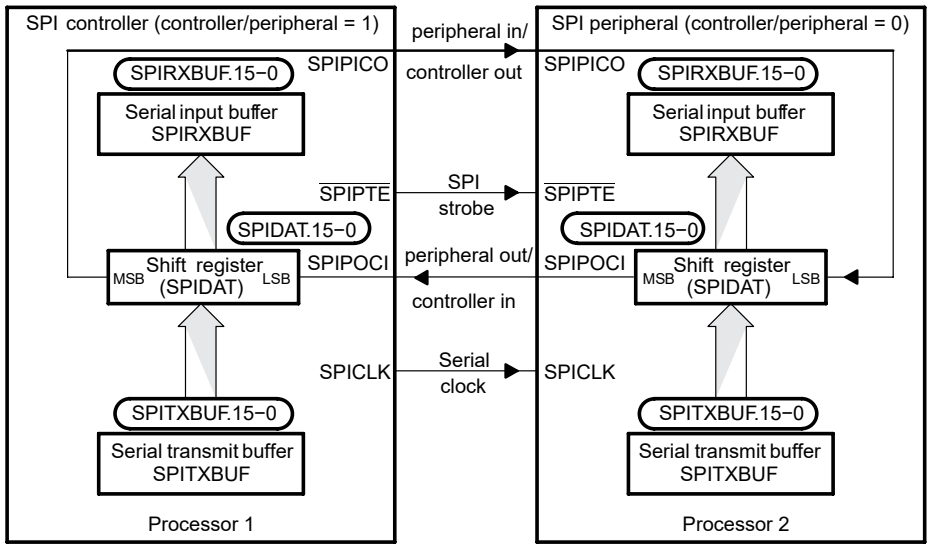


Figure 41-4. SPI Controller/Peripheral Connection

41.3.2 Controller Mode

In controller mode (CONTROLLER_PERIPHERAL = 1), the SPI provides the serial clock on the SPICLK pin for the entire serial communications network. Data is output on the SPIPICO pin and latched from the SPIPOCI pin.

The SPIBRR register determines both the transmit and receive bit transfer rate for the network. SPIBRR can select 125 different data transfer rates.

Data written to SPIDAT or SPITXBUF initiates data transmission on the SPIPICO pin, MSB (most-significant bit) first. Simultaneously, received data is shifted through the SPIPOCI pin into the LSB (least- significant bit) of SPIDAT. When the selected number of bits has been transmitted, the received data is transferred to the SPIRXBUF (buffered receiver) for the CPU to read. Data is stored right-justified in SPIRXBUF.

When the specified number of data bits has been shifted through SPIDAT, the following events occur:

- SPIDAT contents are transferred to SPIRXBUF.
- INT_FLAG bit is set to 1.
- If there is valid data in the transmit buffer SPITXBUF, as indicated by the transmit buffer full flag (BUFFULL_FLAG), this data is transferred to SPIDAT and is transmitted; otherwise, SPICLK stops after all bits have been shifted out of SPIDAT.
- If the SPIINTENA bit is set to 1, an interrupt is asserted.

In a typical application, the $\overline{\text{SPIPTE}}$ pin serves as a chip-enable pin for a SPI peripheral device. This pin is driven low by the controller before transmitting data to the peripheral and is taken high after the transmission is complete.

Figure 41-5 is a block diagram of the SPI in controller mode. The block diagram shows the basic control blocks available in SPI controller mode.

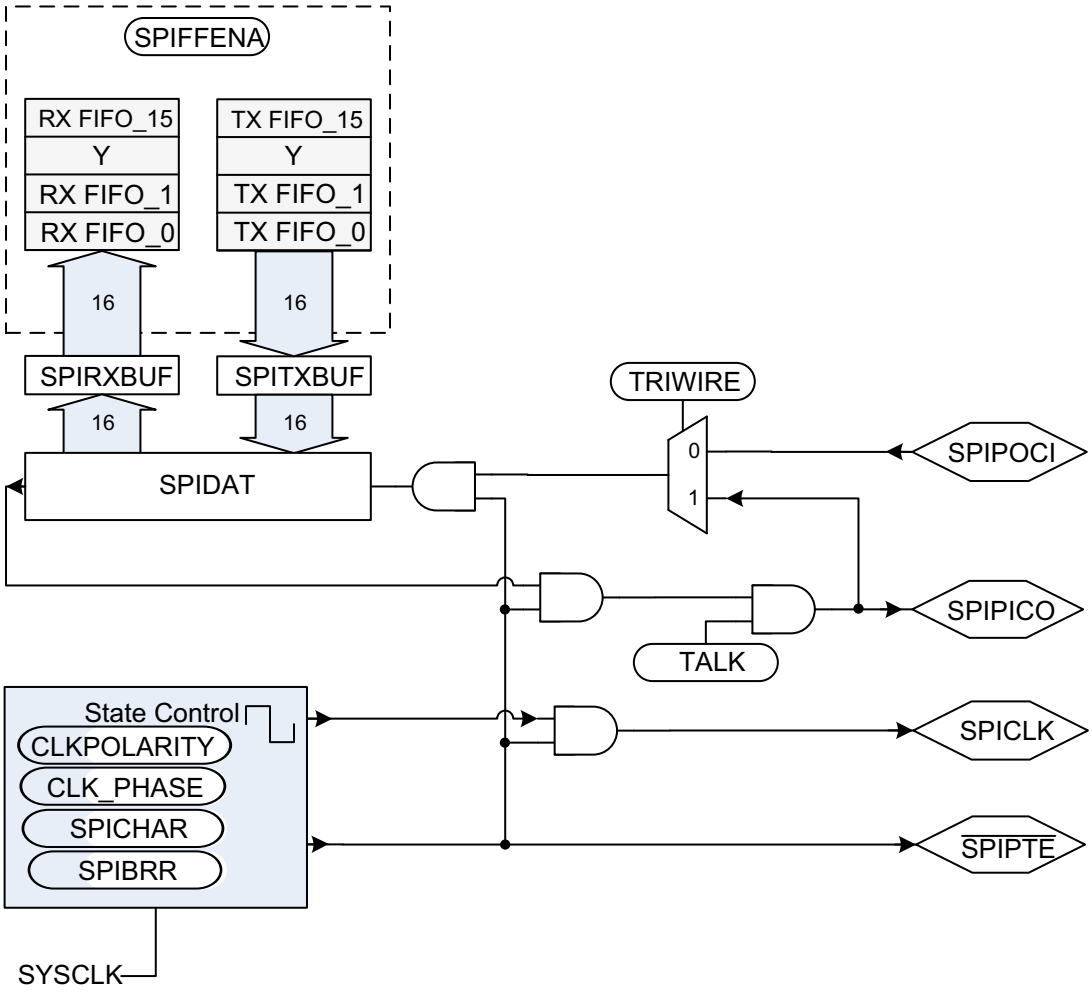


Figure 41-5. SPI Module Controller Configuration

41.3.3 Peripheral Mode

In peripheral mode (CONTROLLER_PERIPHERAL = 0), data shifts out on the SPIPOCI pin and in on the SPIPICO pin. The SPICLK pin is used as the input for the serial shift clock, which is supplied from the external network controller. The transfer rate is defined by this clock. The SPICLK input frequency can be no greater than the SYSCLK frequency divided by 4.

Data written to SPIDAT or SPITXBUF is transmitted to the network when appropriate edges of the SPICLK signal are received from the network controller. A character written to the SPITXBUF register is copied to the SPIDAT register when all bits of the current character in SPIDAT have been shifted out. If no character was previously copied to SPIDAT, then any character written to SPITXBUF is immediately copied to SPIDAT. If a character was previously copied to SPIDAT, any data written to SPITXBUF is not copied to SPIDAT until the current character in SPIDAT has been shifted out. To receive data, the SPI waits for the network controller to send the SPICLK signal and then shifts the data on the SPIPICO pin into SPIDAT. If data is to be transmitted by the peripheral simultaneously, and SPIDAT has not been previously loaded, the character must be written to SPITXBUF before the beginning of the SPICLK signal.

When the TALK bit is cleared, data transmission is disabled, and the output line (SPIPOCI) is put into the high-impedance state. If this occurs while a transmission is active, the current character is completely transmitted even though SPIPOCI is forced into the high-impedance state. This makes sure that the SPI is still able to

receive incoming data correctly. This TALK bit allows many peripheral devices to be tied together on the network, but only one peripheral at a time is allowed to drive the SPIPOCI line.

The $\overline{\text{SPIPTE}}$ pin operates as the peripheral-select pin. An active-low signal on the $\overline{\text{SPIPTE}}$ pin allows the peripheral SPI to transfer data to the serial data line; an inactive-high signal causes the peripheral SPI serial shift register to stop and the serial output pin to be put into the high-impedance state. This allows many peripheral devices to be tied together on the network, although only one peripheral device is selected at a time.

Figure 41-6 is a block diagram of the SPI in peripheral mode. The block diagram shows the basic control blocks available in SPI peripheral mode.

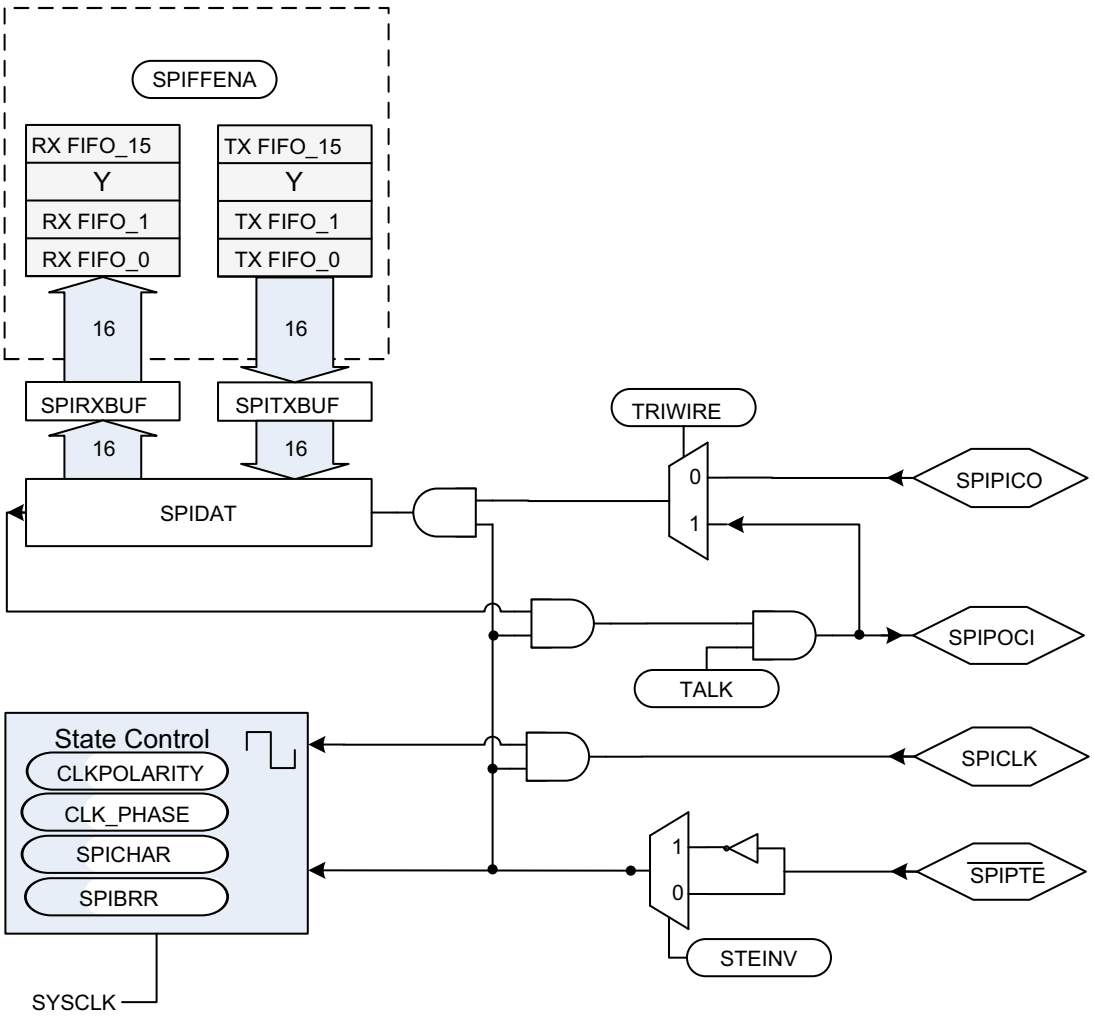


Figure 41-6. SPI Module Peripheral Configuration

41.3.4 Data Format

The 4-bit SPICHAR register field specifies the number of bits in the data character (1 to 16). This information directs the state control logic to count the number of bits received or transmitted to determine when a complete character has been processed.

The following statements apply to characters with fewer than 16 bits:

- Data must be left-justified when written to SPIDAT and SPITXBUF.
- Data read back from SPIRXBUF is right-justified.
- SPIRXBUF contains the most recently received character, right-justified, plus any bits that remain from previous transmissions that have been shifted to the left (shown in Example 41-1).

Example 41-1. Transmission of Bit from SPIRXBUF

Conditions:

1. Transmission character length = 1 bit (specified in SPICHAR bits)
2. The current value of SPIDAT = 737Bh

SPIDAT (before transmission)																	
	0	1	1	1	0	0	1	1	0	1	1	1	1	0	1	1	
SPIDAT (after transmission)																	
(TXed) 0 ←	1	1	1	0	0	1	1	0	1	1	1	1	0	1	1	x(1)	← (RXed)
SPIRXBUF (after transmission)																	
	1	1	1	0	0	1	1	0	1	1	1	1	0	1	1	x(1)	

(1) x = 1, if SPIPOCI data is high; x = 0, if SPIPOCI data is low; controller mode is assumed.

41.3.5 Baud Rate Selection

The SPI module supports 125 different baud rates and four different clock schemes. Depending on whether the SPI clock is in peripheral or controller mode, the SPICLK pin can receive an external SPI clock signal or provide the SPI clock signal, respectively.

- In the peripheral mode, the SPI clock is received on the SPICLK pin from the external source and can be no greater than the SYSCLK frequency divided by 4.
- In the controller mode, the SPI clock is generated by the SPI and is output on the SPICLK pin and can be no greater than the SYSCLK frequency divided by 4.

Note

The baud rate must be configured to not exceed the maximum rated GPIO toggle frequency. Refer to the device data sheet for the maximum GPIO toggle frequency.

Example 41-2 shows how to determine the SPI baud rates.

Example 41-3 shows how to calculate the baud rate of the SPI module in standard SPI mode (HS_MODE = 0).

Example 41-2. Baud Rate Determination

For SPIBRR = 3 to 127:

$$\text{SPI Baud Rate} = \frac{\text{SYSCLK}}{(\text{SPIBRR} + 1)}$$

For SPIBRR = 0, 1, or 2:

$$\text{SPI Baud Rate} = \frac{\text{SYSCLK}}{4}$$

where:

SYSCLK = System clock of the device

SPIBRR = Contents of the SPIBRR in the controller SPI device

To determine what value to load into SPIBRR, you must know the device system clock (SYSCLK) frequency (that is device-specific) and the baud rate at which you are operating.

Example 41-3. Baud Rate Calculation in Non-High Speed Mode (HS_MODE = 0)

$$SPI\ Baud\ Rate = \frac{SYSCLK}{SPIBRR + 1'} \quad SYSCLK = 50\ MHz$$
$$= \frac{50 \times 10^6}{3 + 1}$$
$$= 12.5\ Mbps$$

Example 41-4. Baud Rate Calculation

$$\text{Maximum SPI Baud Rate} = \frac{\text{SYSCLK}}{4}$$
$$= \frac{40 \times 10^6}{4}$$
$$= 10 \times 10^6 \text{ bps}$$

41.3.6 SPI Clocking Schemes

The clock polarity select bit (CLKPOLARITY) and the clock phase select bit (CLK_PHASE) control four different clocking schemes on the SPICLK pin. CLKPOLARITY selects the active edge, either rising or falling, of the clock. CLK_PHASE selects a half-cycle delay of the clock. The four different clocking schemes are:

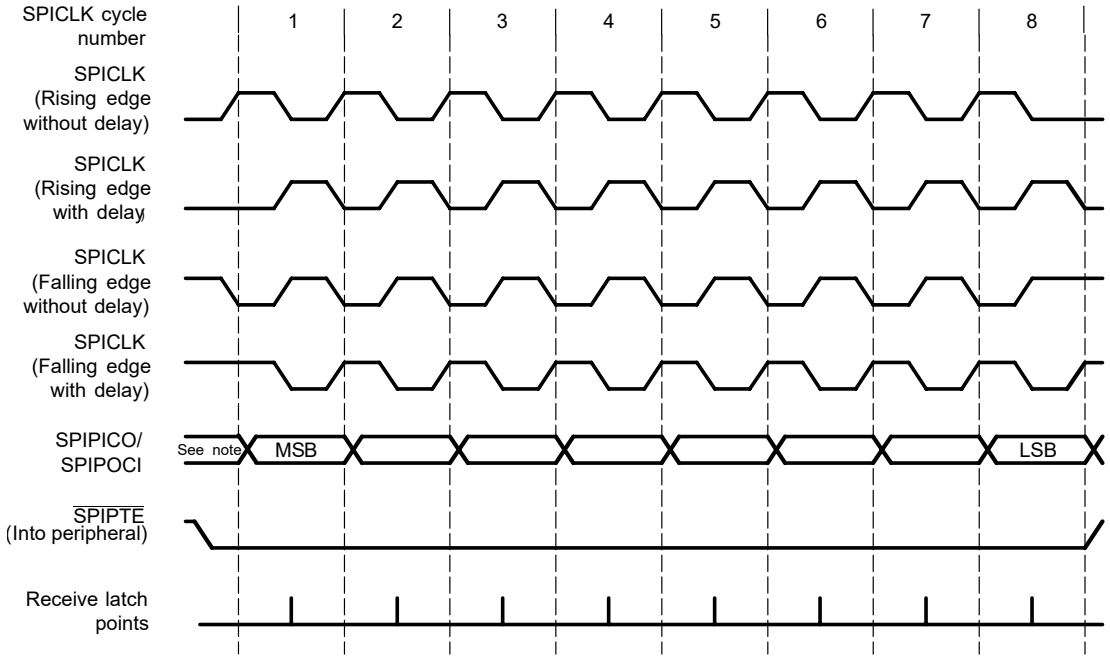
- Falling Edge Without Delay. The SPI transmits data on the falling edge of the SPICLK and receives data on the rising edge of the SPICLK.
- Falling Edge With Delay. The SPI transmits data one half-cycle ahead of the falling edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
- Rising Edge Without Delay. The SPI transmits data on the rising edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
- Rising Edge With Delay. The SPI transmits data one half-cycle ahead of the rising edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal.

The selection procedure for the SPI clocking scheme is shown in Table 41-3. Examples of these four clocking schemes relative to transmitted and received data are shown in Figure 41-7.

Table 41-3. SPI Clocking Scheme Selection Guide

SPICLK Scheme	CLKPOLARITY(1)	CLK_PHASE(1)
Rising edge without delay	0	0
Rising edge with delay	0	1
Falling edge without delay	1	0
Falling edge with delay	1	1

(1) The description of CLK_PHASE and CLKPOLARITY differs between manufacturers. For proper operation, select the desired waveform to determine the clock phase and clock polarity settings.



Note: Previous data bit

Figure 41-7. SPICLK Signal Options

SPICLK symmetry is retained only when the result of (SPIBRR + 1) is an even value. When (SPIBRR + 1) is an odd value and SPIBRR is greater than 3, SPICLK becomes asymmetrical. The low pulse of SPICLK is one SYSCLK cycle longer than the high pulse when CLKPOLARITY bit is clear (0). When CLKPOLARITY bit is set to 1, the high pulse of the SPICLK is one SYSCLK cycle longer than the low pulse, as shown in Figure 41-8.

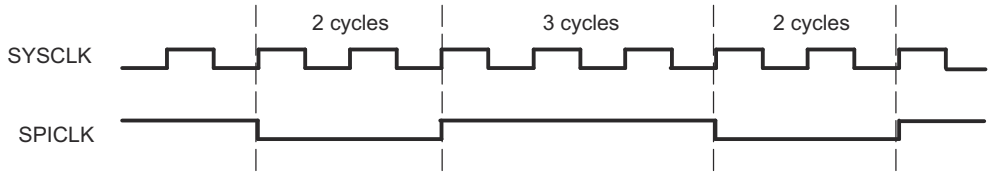


Figure 41-8. SPI: SPICLK-SYSCLK Characteristic when (BRR + 1) is Odd, BRR > 3, and CLKPOLARITY = 1

41.3.7 SPI FIFO Description

The following steps explain the FIFO features and help with programming the SPI FIFOs:

1. **Reset.** At reset the SPI powers up in standard SPI mode and the FIFO function is disabled. The FIFO registers SPIFFTX, SPIFFRX and SPIFFCT remain inactive.
2. **Standard SPI.** The standard 28x SPI mode works with SPIINT/SPIRXINT as the interrupt source.
3. **Mode change.** FIFO mode is enabled by setting the SPIFFENA bit to 1 in the SPIFFTX register. SPIRST can reset the FIFO mode at any stage of the operation.
4. **Active registers.** All the SPI registers and SPI FIFO registers SPIFFTX, SPIFFRX, and SPIFFCT are active.
5. **Interrupts.** FIFO mode has two interrupts: one for the transmit FIFO, SPITXINT; one for the receive FIFO, SPIRXINT. SPIRXINT is the common interrupt for SPI FIFO receive, receive error and receive FIFO overflow conditions. The single SPIINT for both transmit and receive sections of the standard SPI are disabled and this interrupt is serviced as SPI receive FIFO interrupt. For more information, refer to Section 41.2.3.
6. **Buffers.** Transmit and receive buffers are each supplemented with a 16-word FIFO. The one-word transmit buffer (SPITXBUF) of the standard SPI functions as a transition buffer between the transmit FIFO and shift register. The one-word transmit buffer is loaded from transmit FIFO only after the last bit of the shift register is shifted out.
7. **Delayed transfer.** The rate at which transmit words in the FIFO are transferred to transmit shift register is programmable. The SPIFFCT register bits (7−0) FFTXDLY7−FFTXDLY0 define the delay between the word transfer. The delay is defined in number SPI serial clock cycles. The 8-bit register can define a minimum delay of 0 SPICLK cycles and a maximum of 255 SPICLK cycles. With zero delay, the SPI module can transmit data in continuous mode with the FIFO words shifting out back to back. With the 255 clock delay, the SPI module can transmit data in a maximum delayed mode with the FIFO words shifting out with a delay of 255 SPICLK cycles between each words. The programmable delay facilitates glueless interface to various slow SPI peripherals, such as EEPROMs, ADC, DAC, and so on.
8. **FIFO status bits.** Both transmit and receive FIFOs have status bits TXFFST or RXFFST that define the number of words available in the FIFOs at any time. The transmit FIFO reset bit (TXFIFO) and receive reset bit (RXFIFO) reset the FIFO pointers to zero when these bits are set to 1. The FIFOs resume operation from start once these bits are cleared to zero.
9. **Programmable interrupt levels.** Both transmit and receive FIFOs can generate CPU interrupts and RTDMA triggers. The transmit interrupt (SPITXINT) is generated whenever the transmit FIFO status bits (TXFFST) match (less than or equal to) the interrupt trigger level bits (TXFFIL). The receive interrupt (SPIRXINT) is generated whenever the receive FIFO status bits (RXFFST) match (greater than or equal to) the interrupt trigger level RXFFIL. This provides a programmable interrupt trigger for transmit and receive sections of the SPI. The default value for these trigger level bits is 0x11111 for receive FIFO and 0x00000 for transmit FIFO, respectively.

41.3.8 SPI RTDMA Transfers

41.3.8.1 Transmitting Data Using SPI with RTDMA

When using the RTDMA with the TX FIFO, the RTDMA Burst Size (DMA_BURST_SIZE) must be no greater than 16 - TXFFIL, to prevent the RTDMA from writing to an already full FIFO. This leads to data loss and is not recommended.

For complete data transmission, follow these steps:

1. Calculate the total number of 8-bit words to be transmitted. [NUM_WORDS]
2. Decide the transmit FIFO level. [TXFFIL]
3. Calculate the total number of RTDMA transfers. [DMA_TRANSFER_SIZE]
4. Calculate the size of the RTDMA Burst. [DMA_BURST_SIZE]
5. Configure RTDMA using calculated values.
6. Configure SPI with FIFO using the calculated values.

To transfer 128 words to SPI using the RTDMA:

NUM_WORDS: 128

TXFFIL: 8

DMA_TRANSFER_SIZE: (NUM_WORDS /(16 - TXFFIL)) - 1 = (128/8) - 1 = 15 (16 transfers)

DMA_BURST_SIZE: 2(16 - TXFFIL) - 1 = 2(16 - 8) - 1 = 15 (16 words per burst)

Note

Avoid setting TXFFIL to 0h or 10h to make sure of proper RTDMA configuration.

41.3.8.2 Receiving Data Using SPI with RTDMA

When using the RTDMA with the RX FIFO, the RTDMA Burst Size (BURST_SIZE) must be no greater than RXFFIL to prevent the RTDMA from reading from an empty FIFO. To make sure that the RTDMA correctly receives all data from the RX FIFO, the RTDMA Burst Size can equal RXFFIL and also be an integer divisor of the total number of SPI transmissions.

For complete data reception, follow these steps:

1. Calculate the total number of 8-bit words to be received. [NUM_WORDS]
2. Calculate the necessary FIFO level [RXFFIL]
3. Calculate the total number of RTDMA transfers. [DMA_TRANSFER_SIZE]
4. Calculate the size of the RTDMA Burst. [DMA_BURST_SIZE]
5. Configure RTDMA using the calculated values.
6. Configure SPI with FIFO using the calculated values.

To receive 200 words from SPI using the RTDMA:

NUM_WORDS = 200

RXFFIL: 4

DMA_TRANSFER_SIZE: (NUM_WORDS /RXFFIL) - 1 = (200/4) - 1 = 49 (50 transfers)

DMA_BURST_SIZE: 2(RXFFIL) - 1 = 2(4) - 1 = 7 (8 words per burst)

Note

Avoid setting RXFFIL to 0h to make sure proper RTDMA configuration.

41.3.9 SPI High-Speed Mode

The SPI module is capable of reaching full-duplex communication speeds up to SYSCLK/4. For the maximum rated speed, refer to the device data sheet.

To achieve the maximum full-duplex speeds, the following restrictions are placed on the design:

- Single controller to single peripheral configuration is supported.
- Loading on the pins must not exceed the value stated in the device data sheet.

When configuring the GPIOs to support high-speed mode, refer to Section 41.2.2.1 for more information.

41.3.10 SPI 3-Wire Mode Description

SPI 3-wire mode allows for SPI communication over three pins instead of the normal four pins.

In controller mode, if the TRIWIRE bit is set, enabling 3-wire SPI mode, SPIPICOx becomes the bi-directional SPICOCIx (SPI controller out, controller in) pin, and SPIPOCIx is no longer used by the SPI. In peripheral mode, if the TRIWIRE bit is set, SPIPOCIx becomes the bi-directional SPIPIPOx (SPI peripheral in, peripheral out) pin, and SPIPICOx is no longer used by the SPI.

Table 41-4 indicates the pin function differences between 3-wire and 4-wire SPI mode for a controller and peripheral SPI.

Table 41-4. 4-wire versus 3-wire SPI Pin Functions

4-wire SPI	3-wire SPI (Controller)	3-wire SPI (Peripheral)
SPICLKx	SPICLKx	SPICLKx
SPISTEx	SPISTEx	SPISTEx
SPIPICOx	SPICOCIx	Free
SPIPOCIx	Free	SPIPIPOx

Because in 3-wire mode, the receive and transmit paths within the SPI are connected, any data transmitted by the SPI module is also received. The application software must take care to perform a dummy read to clear the SPI data register of the additional received data.

The TALK bit plays an important role in 3-wire SPI mode. The bit must be set to transmit data and cleared prior to reading data. In controller mode, to initiate a read, the application software must write dummy data to the SPI data register (SPIDAT or SPIRXBUF) while the TALK bit is cleared (no data is transmitted out the SPICOCI pin) before reading from the data register.

Figure 41-9 and Figure 41-10 illustrate 3-wire controller and peripheral mode.

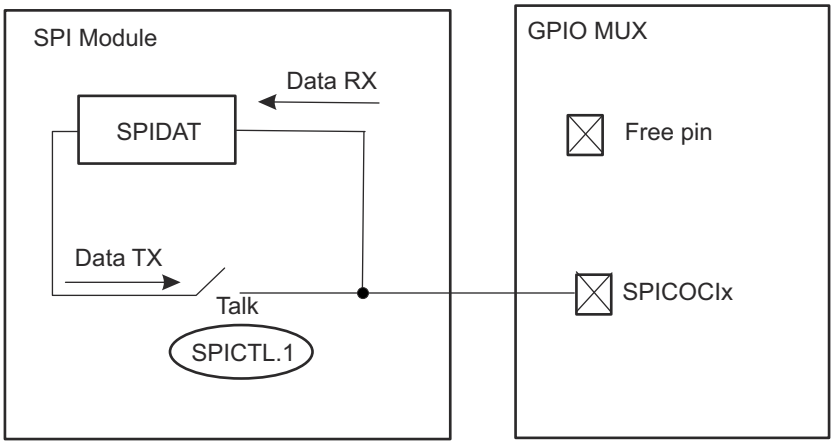


Figure 41-9. SPI 3-wire Controller Mode

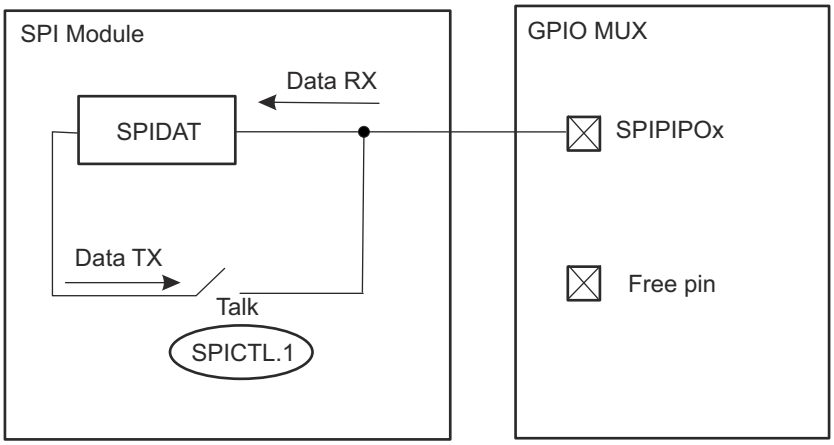


Figure 41-10. SPI 3-wire Peripheral Mode

Table 41-5 indicates how data is received or transmitted in the various SPI modes while the TALK bit is set or cleared.

Table 41-5. 3-Wire SPI Pin Configuration

Pin Mode	SPIPRI[TRIWIRE]	SPICTL[TALK]	SPIPICO	SPIPOCI
		Controller Mode		
4-wire	0	X	TX	RX
3-pin mode	1	0	RX	Disconnect from SPI
		1	TX/RX	
		Peripheral Mode		
4-wire	0	X	RX	TX
3-pin mode	1	0	Disconnect from SPI	RX
		1		TX/RX

41.4 Programming Procedure

This section describes the procedure for configuring the SPI for the various modes of operation.

41.4.1 Initialization Upon Reset

A system reset forces the SPI peripheral into the following default configuration:

- Unit is configured as a peripheral module (CONTROLLER_PERIPHERAL = 0)
- Transmit capability is disabled (TALK = 0)
- Data is latched at the input on the falling edge of the SPICLK signal
- Character length is assumed to be one bit
- SPI interrupts are disabled
- Data in SPIDAT is reset to 0000h

41.4.2 Configuring the SPI

This section describes the procedure in which to configure the SPI module for operation. To prevent unwanted and unforeseen events from occurring during or as a result of initialization changes, clear the SPISWRESET bit before making initialization changes, and then set this bit after initialization is complete. While the SPI is held in reset (SPISWRESET = 0), configuration can be changed in any order. The following list shows the SPI configuration procedure in a logical order. However, the SPI registers can be written with single 16-bit writes, so the order is not required with the exception of SPISWRESET.

Note

Do not change the SPI configuration when communication is in progress.

To change the SPI configuration:

1. Clear the SPI Software Reset bit (SPISWRESET) to 0 to force the SPI to the reset state.
2. Configure the SPI as desired:
   - Select either controller or peripheral mode (CONTROLLER_PERIPHERAL).
   - Choose SPICLK polarity and phase (CLKPOLARITY and CLK_PHASE).
   - Set the desired baud rate (SPIBRR).
   - Set the SPI character length (SPICHAR).
   - Clear the SPI Flags (OVERRUN_FLAG, INT_FLAG).
   - Enable $\overline{\text{SPIPTE}}$ inversion (STEINV), if needed.
   - Enable 3-wire mode (TRIWIRE), if needed.
   - If using FIFO enhancements:
     - Enable the FIFO enhancements (SPIFFENA).
     - Clear the FIFO Flags (TXFFINTCLR, RXFFOVFCLR, and RXFFINTCLR).
     - Release transmit and receive FIFO resets (TXFIFO and RXFIFORESET).
     - Release SPI FIFO channels from reset (SPIRST).
3. If interrupts are used:
   - In non-FIFO mode, enable the receiver overrun and/or SPI interrupts (OVERRUNINTENA and SPIINTENA).
   - In FIFO mode, set the transmit and receive interrupt levels (TXFFIL and RXFFIL) then enable the interrupts (TXFFIENA and RXFFIENA).
4. Set SPISWRESET to 1 to release the SPI from the reset state.

41.4.3 Configuring the SPI for High-Speed Mode

To achieve the maximum rated speeds, the following settings must be made. This example assumes that the device is operating at 100MHz.

Select the appropriate Pin Mux options in GPIO_CTRL_REGS.

During the SPI configuration procedure:

Set HS_MODE to 1.

```
SpiaRegs.SPICCR.bit.HS_MODE = 0x1;
```

Set SPIBRR to 3. SPICLK = SYSCLK/(SPIBRR+1) = 25MHz

```
SpiaRegs.SPIBRR = 0x3;
```

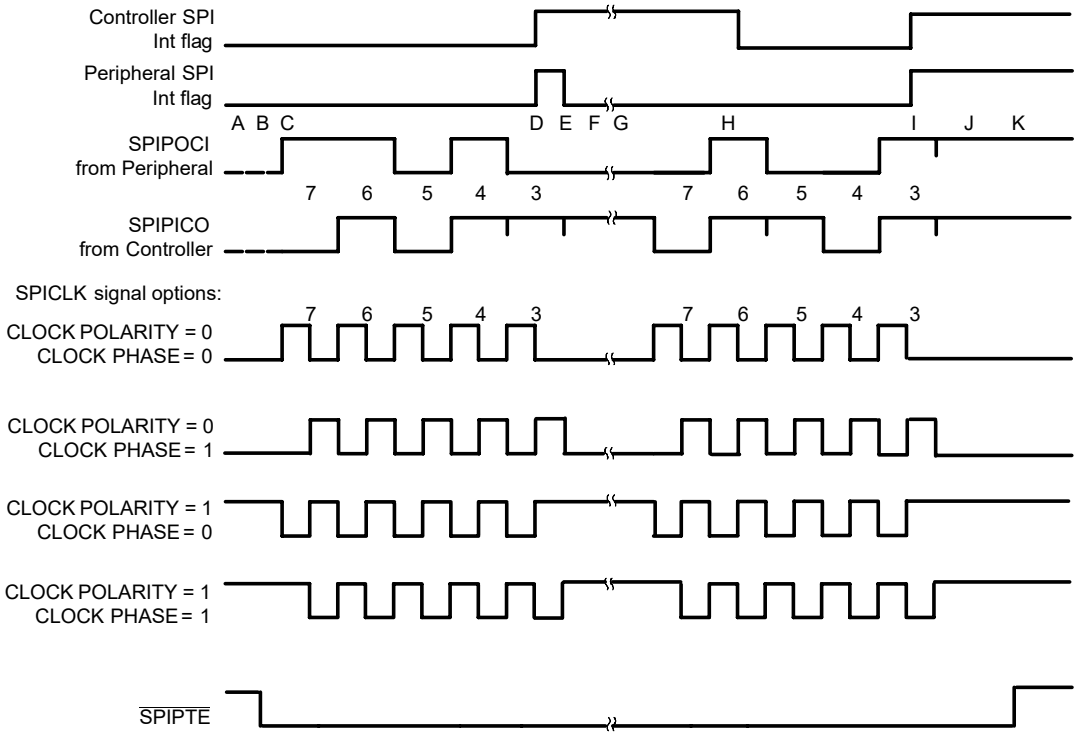
There are no other differences in the configuration from normal SPI operation. Sending and receiving data, RTDMA operation, and interrupts operates without change.

41.4.4 Data Transfer Example

The timing diagram shown in Figure 41-11 shows an SPI data transfer between two devices using a character length of five bits with the SPICLK being symmetrical.

The timing diagram with SPICLK asymmetrical (Figure 41-8) shares similar characterizations with Figure 41-11 except that the data transfer is one SYSCLK cycle longer per bit during the low pulse (CLKPOLARITY = 0) or during the high pulse (CLKPOLARITY = 1) of the SPICLK.

Figure 41-11 is applicable for 8-bit SPI only and is not for C28x devices that are capable of working with 16-bit data. The figure is shown for illustrative purposes only.



A. Peripheral writes 0D0h to SPIDAT and waits for the controller to shift out the data.
B. Controller sets the peripheral $\overline{\text{SPIPTE}}$ signal low (active).
C. Controller writes 058h to SPIDAT, which starts the transmission procedure.
D. First byte is finished and sets the interrupt flags.
E. Peripheral reads 0Bh from the SPIRXBUF (right-justified).
F. Peripheral writes 04Ch to SPIDAT and waits for the controller to shift out the data.
G. Controller writes 06Ch to SPIDAT, which starts the transmission procedure.
H. Controller reads 01Ah from the SPIRXBUF (right-justified).
I. Second byte is finished and sets the interrupt flags.
J. Controller reads 89h and the peripheral reads 8Dh from the respective SPIRXBUF. After the user software masks off the unused bits, the controller receives 09h and the peripheral receives 0Dh.
K. Controller clears the peripheral $\overline{\text{SPIPTE}}$ signal high (inactive).

Figure 41-11. Five Bits per Character

41.4.5 SPI 3-Wire Mode Code Examples

In addition to the normal SPI initialization, to configure the SPI module for 3-wire mode, the TRIWIRE bit (SPIPRI.0) must be set to 1. After initialization, there are several considerations to take into account when transmitting and receiving data in 3-wire controller and peripheral mode. The following examples demonstrate these considerations.

In 3-wire controller mode, SPICLKx, $\overline{\text{SPIPTE}}$x, and SPIPICOx pins must be configured as SPI pins (SPIPOCIx pin can be configured as non-SPI pin). When the controller transmits, the controller receives the data the controller transmits (because SPIPICOx and SPIPOCIx are connected internally in 3-wire mode). Therefore, the junk data received must be cleared from the receive buffer every time data is transmitted.

Example 41-5. 3-Wire Controller Mode Transmit

```
Uint16 data;
Uint16 dummy;
     SpiaRegs.SPICTL.bit.TALK = 1;              // Enable Transmit path
     SpiaRegs.SPITXBUF = data; // Controller transmits data
     while(SpiaRegs.SPISTS.bit.INT_FLAG !=1) {} // Waits until data rx'd
     dummy = SpiaRegs.SPIRXBUF;                 // Clears junk data because
                                                // rx'd same data as tx'd
```

To receive data in 3-wire controller mode, the controller must clear the TALK (SPICTL.1) bit to 0 to close the transmit path and then transmit dummy data to initiate the transfer from the peripheral. Because the TALK bit is 0, unlike in transmit mode, the controller dummy data does not appear on the SPIPICOx pin, and the controller does not receive the dummy data. Instead, the data from the peripheral is received by the controller.

Example 41-6. 3-Wire Controller Mode Receive

```
Uint16 rdata;
Uint16 dummy;
     SpiaRegs.SPICTL.bit.TALK = 0;              // Disable Transmit path
     SpiaRegs.SPITXBUF = dummy;                 // Send dummy to start tx
     // NOTE: because TALK = 0, data does not tx onto SPIPICOA pin
     while(SpiaRegs.SPISTS.bit.INT_FLAG !=1) {} // Wait until data received
     rdata = SpiaRegs.SPIRXBUF;                 // Controller reads data
```

In 3-wire peripheral mode, SPICLKx, SPISTEx, and SPIPOCIx pins must be configured as SPI pins (SPIPICOx pin can be configured as non-SPI pin). Like in controller mode, when transmitting, the peripheral receives the data transmitted and must clear this junk data from the receive buffer.

Example 41-7. 3-Wire Peripheral Mode Transmit

```
Uint16 data;
Uint16 dummy;
     SpiaRegs.SPICTL.bit.TALK = 1;              // Enable Transmit path
     SpiaRegs.SPITXBUF = data;                  // Peripheral transmits data
     while(SpiaRegs.SPISTS.bit.INT_FLAG !=1) {} // Wait until data rx'd
     dummy = SpiaRegs.SPIRXBUF;                 // Clears junk data
```

As in 3-wire controller mode, the TALK bit must be cleared to 0. Otherwise, the peripheral receives data normally.

Example 41-8. 3-Wire Peripheral Mode Receive

```
Uint16 rdata;
    SpiaRegs.SPICTL.bit.TALK = 0;              // Disable Transmit path
    while(SpiaRegs.SPISTS.bit.INT_FLAG !=1) {} // Waits until data rx'd
    rdata = SpiaRegs.SPIRXBUF;                 // Peripheral reads data
```

41.4.6 SPI STEINV Bit in Digital Audio Transfers

On those devices with two SPI modules, enabling the STEINV bit on one of the SPI modules allows the pair of SPIs to receive both left and right-channel digital audio data in peripheral mode. The SPI module that receives a normal active-low $\overline{\text{SPIPTE}}$ signal stores right-channel data, and the SPI module that receives an inverted active-high $\overline{\text{SPIPTE}}$ signal stores left-channel data from the controller. To receive digital audio data from a digital audio interface receiver, the SPI modules can be connected as shown in Figure 41-12.

Note

This configuration is only applicable to peripheral mode (CONTROLLER_PERIPHERAL = 0). When the SPI is configured as controller (CONTROLLER_PERIPHERAL = 1), the STEINV bit has no effect on the $\overline{\text{SPIPTE}}$ pin.

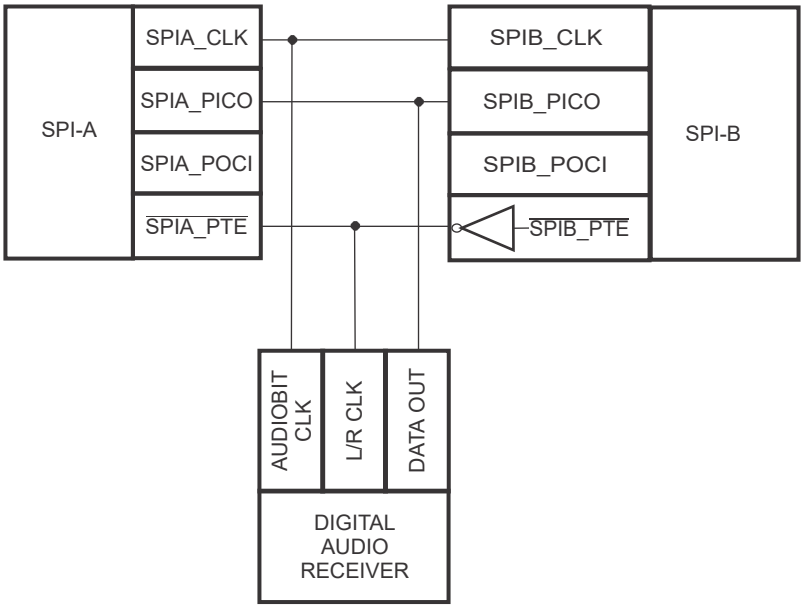


Figure 41-12. SPI Digital Audio Receiver Configuration Using Two SPIs

Standard C29x SPI timing requirements limit the number of digital audio interface formats supported using the 2-SPI configuration with the STEINV bit. See the device data sheet electrical specifications for SPI timing requirements. With the SPI clock phase configured such that the CLKPOLARITY bit is 0 and the CLK_PHASE bit is 1 (data latched on rising edge of clock), standard right-justified digital audio interface data format is supported as shown in Figure 41-13.

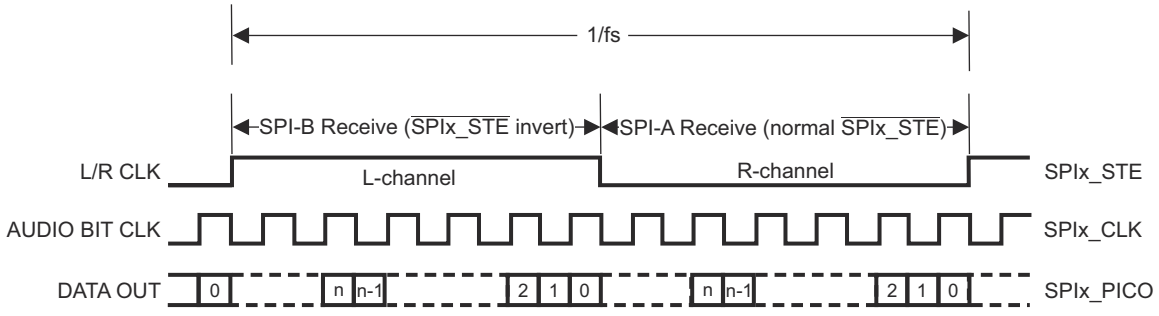


Figure 41-13. Standard Right-Justified Digital Audio Data Format

41.5 Software

41.5.1 SPI Registers to Driverlib Functions

Table 41-6. SPI Registers to Driverlib Functions

File	Driverlib Function
SPICCR	
spi.c	SPI_setConfig
spi.c	SPI_clearInterruptStatus
spi.c	SPI_pollingNonFIFOTransaction
spi.c	SPI_pollingFIFOTransaction
spi.h	SPI_enableModule
spi.h	SPI_disableModule
spi.h	SPI_setcharLength
spi.h	SPI_enableLoopback
spi.h	SPI_disableLoopback
spi.h	SPI_enableHighSpeedMode
spi.h	SPI_disableHighSpeedMode
SPICTL	
spi.c	SPI_setConfig
spi.c	SPI_enableInterrupt
spi.c	SPI_disableInterrupt
spi.h	SPI_enableTalk
spi.h	SPI_disableTalk
SPISTS	
spi.c	SPI_getInterruptStatus
spi.c	SPI_clearInterruptStatus
spi.h	SPI_writeDataBlockingNonFIFO
spi.h	SPI_readDataBlockingNonFIFO
SPIBRR	
spi.c	SPI_setConfig
spi.c	SPI_setBaudRate
SPIRXEMU	
spi.h	SPI_readRxEmulationBuffer
SPIRXBUF	
spi.h	SPI_readDataNonBlocking
spi.h	SPI_readDataBlockingFIFO
spi.h	SPI_readDataBlockingNonFIFO
SPITXBUF	
spi.h	SPI_writeDataNonBlocking
spi.h	SPI_writeDataBlockingFIFO
spi.h	SPI_writeDataBlockingNonFIFO
SPIDAT	
-	
SPIFFTX	
spi.c	SPI_enableInterrupt
spi.c	SPI_disableInterrupt
spi.c	SPI_getInterruptStatus

Table 41-6. SPI Registers to Driverlib Functions (continued)

File	Driverlib Function
spi.c	SPI_clearInterruptStatus
spi.h	SPI_enableFIFO
spi.h	SPI_disableFIFO
spi.h	SPI_resetTxFIFO
spi.h	SPI_setFIFOInterruptLevel
spi.h	SPI_getFIFOInterruptLevel
spi.h	SPI_getTxFIFOStatus
spi.h	SPI_isBusy
spi.h	SPI_reset
SPIFFRX	
spi.c	SPI_enableInterrupt
spi.c	SPI_disableInterrupt
spi.c	SPI_getInterruptStatus
spi.c	SPI_clearInterruptStatus
spi.h	SPI_enableFIFO
spi.h	SPI_disableFIFO
spi.h	SPI_resetRxFIFO
spi.h	SPI_setFIFOInterruptLevel
spi.h	SPI_getFIFOInterruptLevel
spi.h	SPI_getRxFIFOStatus
SPIFFCT	
spi.h	SPI_setTxFifoTransmitDelay
SPIPRI	
spi.h	SPI_enableTriWire
spi.h	SPI_disableTriWire
spi.h	SPI_setPTESignalPolarity
spi.h	SPI_setEmulationMode

41.5.2 SPI Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/spi

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

41.5.2.1 SPI Digital Loopback - SINGLE_CORE

FILE: spi_ex1_loopback.c

This program uses the internal loopback test mode of the SPI module. This is a very basic loopback that does not use the FIFOs or interrupts. A stream of data is sent and then compared to the received stream. The pinmux and SPI modules are configured through the sysconfig file.

The sent data looks like this:
0000 0001 0002 0003 0004 0005 0006 0007 .... FFFE FFFF 0000

This pattern is repeated forever.

External Connections

- None

Watch Variables

- *sData* - Data to send
- *rData* - Received data

41.5.2.2 SPI Digital Loopback with FIFO Interrupts - SINGLE_CORE

FILE: spi_ex2_loopback_fifo_interrupts.c

This program uses the internal loopback test mode of the SPI module. Both the SPI FIFOs are used, and SPI RX interrupt is used.

A stream of data is sent and then compared to the received stream. The sent data looks like this:
0000 0001
0001 0002
0002 0003
....
FFFE FFFF
FFFF 0000
etc..
This pattern is repeated forever.

Note : The SPI peripheral generates level interrupts, which should be cleared in ISR to avoid generating false pending interrupt on clear edge, followed by some wait cycles

External Connections

- None

Watch Variables

- *sData* - Data to send
- *rData* - Received data
- *rDataPoint* - Used to keep track of the last position in the receive stream for error checking

41.5.2.3 SPI Digital External Loopback without FIFO Interrupts - SINGLE_CORE

FILE: spi_ex3_external_loopback.c

This program uses the external loopback between two SPI modules. Both the SPI FIFOs and interrupts are not used in this example. SPIA is configured as a peripheral and SPI B is configured as controller. This example demonstrates full duplex communication where both controller and peripheral transmits and receives data simultaneously.

External Connections
-GPIO16 and GPIO63 - SPIPICO -GPIO17 and GPIO25 - SPIPOCI -GPIO34 and GPIO26 - SPICLK -GPIO61 and GPIO27 - SPISTE

Watch Variables

- *TxData_SPIA* - Data send from SPIA (peripheral)
- *TxData_SPIB* - Data send from SPIB (controller)
- *RxData_SPIA* - Data received by SPIA (peripheral)
- *RxData_SPIB* - Data received by SPIB (controller)

41.5.2.4 SPI Digital External Loopback with FIFO Interrupts - SINGLE_CORE

FILE: spi_ex4_external_loopback_fifo_interrupts.c

This program uses the external loopback between two SPI modules. Both the SPI FIFOs are used. SPI-A is configured as a peripheral and receives data from SPI-B which is configured as a controller. SPI-A RX interrupt is used.

A stream of data is sent and then compared to the received stream. The sent data looks like this:
0000 0001
0001 0002

0002 0003
....
FFFE FFFF
FFFF 0000
etc..
This pattern is repeated forever.

Note : The SPI peripheral generates level interrupts, which should be cleared in ISR to avoid generating false pending interrupt on clear edge, followed by some wait cycles

External Connections
-GPIO16 and GPIO63 - SPIPICO -GPIO17 and GPIO25 - SPIPOCI -GPIO34 and GPIO26 - SPICLK -GPIO61 and GPIO27 - SPISTE

Watch Variables
- *sData* - Data to send
- *rData* - Received data
- *rDataPoint* - Used to keep track of the last position in the receive stream for error checking

41.5.2.5 SPI Digital Loopback with DMA - SINGLE_CORE

FILE: spi_ex5_loopback_dma.c

This program uses the internal loopback test mode of the SPI module. Both DMA interrupts and the SPI FIFOs are used. When the SPI transmit FIFO has enough space (as indicated by its FIFO level interrupt signal), the DMA will transfer data from global variable sData into the FIFO. This will be transmitted to the receive FIFO via the internal loopback.

When enough data has been placed in the receive FIFO (as indicated by its FIFO level interrupt signal), the DMA will transfer the data from the FIFO into global variable rData.

When all data has been placed into rData, a check of the validity of the data will be performed in one of the DMA channels' ISRs.

External Connections
- None

Watch Variables
- *sData* - Data to send
- *rData* - Received data

41.6 SPI Registers

This Section describes the SPI Registers.

41.6.1 SPI Base Address Table

Table 41-7. SPI Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
SPI_REGS	SPIA_BASE	0x7015_8000	YES	YES	YES	YES	YES	YES	-	YES
SPI_REGS	SPIB_BASE	0x7015_9000	YES	YES	YES	YES	YES	YES	-	YES
SPI_REGS	SPIC_BASE	0x7015_A000	YES	YES	YES	YES	YES	YES	-	YES
SPI_REGS	SPID_BASE	0x7015_B000	YES	YES	YES	YES	YES	YES	-	YES
SPI_REGS	SPIE_BASE	0x7015_C000	YES	YES	YES	YES	YES	YES	-	YES

41.6.2 SPI_REGS Registers

Table 41-8 lists the memory-mapped registers for the SPI_REGS registers. All register offset addresses not listed in Table 41-8 should be considered as reserved locations and the register contents should not be modified.

Table 41-8. SPI_REGS Registers

Offset	Acronym	Register Name	Protection
0h	SPICCR	SPI Configuration Control Register	
2h	SPICTL	SPI Operation Control Register	
4h	SPISTS	SPI Status Register	
8h	SPIBRR	SPI Baud Rate Register	
Ch	SPIRXEMU	SPI Emulation Buffer Register	
Eh	SPIRXBUF	SPI Serial Input Buffer Register	
10h	SPITXBUF	SPI Serial Output Buffer Register	
12h	SPIDAT	SPI Serial Data Register	
14h	SPIFFTX	SPI FIFO Transmit Register	
16h	SPIFFRX	SPI FIFO Receive Register	
18h	SPIFFCT	SPI FIFO Control Register	
1Eh	SPIPRI	SPI Priority Control Register	

Complex bit access types are encoded to fit into small table cells. Table 41-9 shows the codes that are used for access types in this section.

Table 41-9. SPI_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
RC	R C	Read to Clear
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
Reset or Default Value		
-n		Value after reset or the default value

41.6.2.1 SPICCR Register (Offset = 0h) [Reset = 0000h]

SPICCR is shown in Figure 41-14 and described in Table 41-10.

Return to the Summary Table.

SPICCR controls the setup of the SPI for operation.

Figure 41-14. SPICCR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
SPISWRESET	CLKPOLARITY	HS_MODE	SPILBK	SPICHAR			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h			

Table 41-10. SPICCR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	SPISWRESET	R/W	0h	SPI Software Reset When changing configuration, you should clear this bit before the changes and set this bit before resuming operation. Reset type: SYSRSn 0h (R/W) = Initializes the SPI operating flags to the reset condition. Specifically, the RECEIVER OVERRUN Flag bit (SPISTS.7), the SPI INT FLAG bit (SPISTS.6), and the TXBUF FULL Flag bit (SPISTS.5) are cleared. SPIPTE will become inactive. SPICLK will be immediately driven to 0 regardless of the clock polarity. The SPI configuration remains unchanged. 1h (R/W) = SPI is ready to transmit or receive the next character. When the SPI SW RESET bit is a 0, a character written to the transmitter will not be shifted out when this bit is set. A new character must be written to the serial data register. SPICLK will be returned to its inactive state one SPICLK cycle after this bit is set.
6	CLKPOLARITY	R/W	0h	Shift Clock Polarity This bit controls the polarity of the SPICLK signal. CLOCK POLARITY and POLARITY CLOCK PHASE (SPICTL.3) control four clocking schemes on the SPICLK pin. Reset type: SYSRSn 0h (R/W) = Data is output on rising edge and input on falling edge. When no SPI data is sent, SPICLK is at low level. The data input and output edges depend on the value of the CLOCK PHASE bit (SPICTL.3) as follows: - CLOCK PHASE = 0: Data is output on the rising edge of the SPICLK signal. Input data is latched on the falling edge of the SPICLK signal. - CLOCK PHASE = 1: Data is output one half-cycle before the first rising edge of the SPICLK signal and on subsequent falling edges of the SPICLK signal. Input data is latched on the rising edge of the SPICLK signal. 1h (R/W) = Data is output on falling edge and input on rising edge. When no SPI data is sent, SPICLK is at high level. The data input and output edges depend on the value of the CLOCK PHASE bit (SPICTL.3) as follows: - CLOCK PHASE = 0: Data is output on the falling edge of the SPICLK signal. Input data is latched on the rising edge of the SPICLK signal. - CLOCK PHASE = 1: Data is output one half-cycle before the first falling edge of the SPICLK signal and on subsequent rising edges of the SPICLK signal. Input data is latched on the falling edge of the SPICLK signal.

Table 41-10. SPICCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	HS_MODE	R/W	0h	High Speed Mode Enable Bits This bit determines if the High Speed mode is enabled. The correct GPIOs should be selected in the GPxGMUX/GPxMUX registers. Reset type: SYSRSn 0h (R/W) = SPI High Speed mode disabled. This is the default value after reset. 1h (R/W) = SPI High Speed mode enabled,
4	SPILBK	R/W	0h	SPI Loopback Mode Select Loopback mode allows module validation during device testing. This mode is valid only in CONTROLLER mode of the SPI. Reset type: SYSRSn 0h (R/W) = SPI loopback mode disabled. This is the default value after reset. 1h (R/W) = SPI loopback mode enabled, PICO/POCI lines are connected internally. Used for module self-tests.
3-0	SPICHAR	R/W	0h	Character Length Control Bits These four bits determine the number of bits to be shifted in or SPI CHAR0 out as a single character during one shift sequence. SPICHAR = Word length - 1 Reset type: SYSRSn 0h (R/W) = 1-bit word 1h (R/W) = 2-bit word 7h (R/W) = 8-bit word Fh (R/W) = 16-bit word

41.6.2.2 SPICTL Register (Offset = 2h) [Reset = 0000h]

SPICTL is shown in Figure 41-15 and described in Table 41-11.

Return to the Summary Table.

SPICTL controls data transmission, the SPI's ability to generate interrupts, the SPICLK phase, and the operational mode (PERIPHERAL or CONTROLLER).

Figure 41-15. SPICTL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			OVERRUNINTENA	CLK_PHASE	CONTROLLER_PERIPHERAL	TALK	SPIINTENA
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 41-11. SPICTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R	0h	Reserved
4	OVERRUNINTENA	R/W	0h	Overrun Interrupt Enable Overrun Interrupt Enable. Setting this bit causes an interrupt to be generated when the RECEIVER OVERRUN Flag bit (SPISTS.7) is set by hardware. Interrupts generated by the RECEIVER OVERRUN Flag bit and the SPI INT FLAG bit (SPISTS.6) share the same interrupt vector. Reset type: SYSRSn 0h (R/W) = Disable RECEIVER OVERRUN interrupts. 1h (R/W) = Enable RECEIVER_OVERRUN interrupts.
3	CLK_PHASE	R/W	0h	SPI Clock Phase Select This bit controls the phase of the SPICLK signal. CLOCK PHASE and CLOCK POLARITY (SPICCR.6) make four different clocking schemes possible (see clocking figures in SPI chapter). When operating with CLOCK PHASE high, the SPI (CONTROLLER or PERIPHERAL) makes the first bit of data available after SPIDAT is written and before the first edge of the SPICLK signal, regardless of which SPI mode is being used. Reset type: SYSRSn 0h (R/W) = Normal SPI clocking scheme, depending on the CLOCK POLARITY bit (SPICCR.6). 1h (R/W) = SPICLK signal delayed by one half-cycle. Polarity determined by the CLOCK POLARITY bit.
2	CONTROLLER_PERIPHERAL	R/W	0h	SPI Network Mode Control This bit determines whether the SPI is a network CONTROLLER or PERIPHERAL. After SPI reset, SPI is automatically configured as a PERIPHERAL Reset type: SYSRSn 0h (R/W) = SPI is configured as a PERIPHERAL. 1h (R/W) = SPI is configured as a CONTROLLER.

Table 41-11. SPICTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	TALK	R/W	0h	Transmit Enable The TALK bit can disable data transmission (CONTROLLER or PERIPHERAL) by placing the serial data output in the high-impedance state. If this bit is disabled during a transmission, the transmit shift register continues to operate until the previous character is shifted out. When the TALK bit is disabled, the SPI is still able to receive characters and update the status flags. TALK is cleared (disabled) by a system reset. Reset type: SYSRSn 0h (R/W) = Disables transmission: - PERIPHERAL mode operation: If not previously configured as a general-purpose I/O pin, the SPIPOCI pin will be put in the high-impedance state. - CONTROLLER mode operation: If not previously configured as a general-purpose I/O pin, the SPIPICO pin will be put in the high-impedance state. 1h (R/W) = Enables transmission For the 4-pin option, ensure to enable the receiver's SPIPTEn input pin.
0	SPIINTENA	R/W	0h	SPI Interrupt Enable This bit controls the SPI's ability to generate a transmit/receive interrupt. The SPI INT FLAG bit (SPISTS.6) is unaffected by this bit. Reset type: SYSRSn 0h (R/W) = Disables the interrupt. 1h (R/W) = Enables the interrupt.

41.6.2.3 SPISTS Register (Offset = 4h) [Reset = 0000h]

SPISTS is shown in Figure 41-16 and described in Table 41-12.

Return to the Summary Table.

SPISTS contrains interrupt and status bits.

Figure 41-16. SPISTS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
OVERRUN_FLAG	INT_FLAG	BUFFULL_FLAG	RESERVED				
W1C-0h	RC-0h	R-0h	R-0h				

Table 41-12. SPISTS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	OVERRUN_FLAG	W1C	0h	SPI Receiver Overrun Flag This bit is a read/clear-only flag. The SPI hardware sets this bit when a receive or transmit operation completes before the previous character has been read from the buffer. The bit is cleared in one of three ways: - Writing a 1 to this bit - Writing a 0 to SPI SW RESET (SPICCR.7) - Resetting the system If the OVERRUN INT ENA bit (SPICTL.4) is set, the SPI requests only one interrupt upon the first occurrence of setting the RECEIVER OVERRUN Flag bit. Subsequent overruns will not request additional interrupts if this flag bit is already set. This means that in order to allow new overrun interrupt requests the user must clear this flag bit by writing a 1 to SPISTS.7 each time an overrun condition occurs. In other words, if the RECEIVER OVERRUN Flag bit is left set (not cleared) by the interrupt service routine, another overrun interrupt will not be immediately re-entered when the interrupt service routine is exited. Reset type: SYSRSn 0h (R/W) = A receive overrun condition has not occurred. 1h (R/W) = The last received character has been overwritten and therefore lost (when the SPIRXBUF was overwritten by the SPI module before the previous character was read by the user application). Writing a '1' will clear this bit. The RECEIVER OVERRUN Flag bit should be cleared during the interrupt service routine because the RECEIVER OVERRUN Flag bit and SPI INT FLAG bit (SPISTS.6) share the same interrupt vector. This will alleviate any possible doubt as to the source of the interrupt when the next byte is received.

Table 41-12. SPISTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	INT_FLAG	RC	0h	SPI Interrupt Flag SPI INT FLAG is a read-only flag. Hardware sets this bit to indicate that the SPI has completed sending or receiving the last bit and is ready to be serviced. This flag causes an interrupt to be requested if the SPI INT ENA bit (SPICTL.0) is set. The received character is placed in the receiver buffer at the same time this bit is set. This bit is cleared in one of three ways: - Reading SPIRXBUF - Writing a 0 to SPI SW RESET (SPICCR.7) - Resetting the system Note: This bit should not be used if FIFO mode is enabled. The internal process of copying the received word from SPIRXBUF to the Receive FIFO will clear this bit. Use the FIFO status, or FIFO interrupt bits for similar functionality. Reset type: SYSRSn 0h (R/W) = No full words have been received or transmitted. 1h (R/W) = Indicates that the SPI has completed sending or receiving the last bit and is ready to be serviced.
5	BUFFULL_FLAG	R	0h	SPI Transmit Buffer Full Flag This read-only bit gets set to 1 when a character is written to the SPI Transmit buffer SPITXBUF. It is cleared when the character is automatically loaded into SPIDAT when the shifting out of a previous character is complete. Reset type: SYSRSn 0h (R/W) = Transmit buffer is not full. 1h (R/W) = Transmit buffer is full.
4-0	RESERVED	R	0h	Reserved

41.6.2.4 SPIBRR Register (Offset = 8h) [Reset = 0000h]

SPIBRR is shown in Figure 41-17 and described in Table 41-13.

Return to the Summary Table.

SPIBRR contains the bits used for baud-rate selection.

Figure 41-17. SPIBRR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	SPI_BIT_RATE						
R-0h	R/W-0h						

Table 41-13. SPIBRR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6-0	SPI_BIT_RATE	R/W	0h	SPI Baud Rate Control These bits determine the bit transfer rate if the SPI is the network SPI BIT RATE 0 CONTROLLER. There are 125 data-transfer rates (each a function of the CPU clock, SYSCLK) that can be selected. One data bit is shifted per SPICLK cycle. (SPICLK is the baud rate clock output on the SPICLK pin.) If the SPI is a network PERIPHERAL, the module receives a clock on the SPICLK pin from the network CONTROLLER. Therefore, these bits have no effect on the SPICLK signal. The frequency of the input clock from the CONTROLLER should not exceed the PERIPHERAL SPI's SYSCLK signal divided by 4. In CONTROLLER mode, the SPI clock is generated by the SPI and is output on the SPICLK pin. The SPI baud rates are determined by the following formula: For SPIBRR = 3 to 127: SPI Baud Rate = SYSCLK / (SPIBRR + 1) For SPIBRR = 0, 1, or 2: SPI Baud Rate = SYSCLK / 4 Reset type: SYSRSn 3h (R/W) = SPI Baud Rate = SYSCLK/4 4h (R/W) = SPI Baud Rate = SYSCLK/5 7Eh (R/W) = SPI Baud Rate = SYSCLK/127 7Fh (R/W) = SPI Baud Rate = SYSCLK/128

41.6.2.5 SPIRXEMU Register (Offset = Ch) [Reset = 0000h]

SPIRXEMU is shown in Figure 41-18 and described in Table 41-14.

Return to the Summary Table.

SPIRXEMU contains the received data. Reading SPIRXEMU does not clear the SPI INT FLAG bit of SPISTS. This is not a real register but a dummy address from which the contents of SPIRXBUF can be read by the emulator without clearing the SPI INT FLAG.

Figure 41-18. SPIRXEMU Register

15	14	13	12	11	10	9	8
ERXBn							
R-0h							
7	6	5	4	3	2	1	0
ERXBn							
R-0h							

Table 41-14. SPIRXEMU Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	ERXBn	R	0h	Emulation Buffer Received Data SPIRXEMU functions almost identically to SPIRXBUF, except that reading SPIRXEMU does not clear the SPI INT FLAG bit (SPISTS.6). Once the SPIDAT has received the complete character, the character is transferred to SPIRXEMU and SPIRXBUF, where it can be read. At the same time, SPI INT FLAG is set. This mirror register was created to support emulation. Reading SPIRXBUF clears the SPI INT FLAG bit (SPISTS.6). In the normal operation of the emulator, the control registers are read to continually update the contents of these registers on the display screen. SPIRXEMU was created so that the emulator can read this register and properly update the contents on the display screen. Reading SPIRXEMU does not clear the SPI INT FLAG bit, but reading SPIRXBUF clears this flag. In other words, SPIRXEMU enables the emulator to emulate the true operation of the SPI more accurately. It is recommended that you view SPIRXEMU in the normal emulator run mode. Reset type: SYSRSn

41.6.2.6 SPIRXBUF Register (Offset = Eh) [Reset = 0000h]

SPIRXBUF is shown in Figure 41-19 and described in Table 41-15.

Return to the Summary Table.

SPIRXBUF contains the received data. Reading SPIRXBUF clears the SPI INT FLAG bit in SPISTS. If FIFO mode is enabled, reading this register will also decrement the RXFFST counter in SPIFFRX.

Figure 41-19. SPIRXBUF Register

15	14	13	12	11	10	9	8
RXBn							
R-0h							
7	6	5	4	3	2	1	0
RXBn							
R-0h							

Table 41-15. SPIRXBUF Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RXBn	R	0h	Received Data Once SPIDAT has received the complete character, the character is transferred to SPIRXBUF, where it can be read. At the same time, the SPI INT FLAG bit (SPISTS.6) is set. Since data is shifted into the SPI's most significant bit first, it is stored right-justified in this register. Reset type: SYSRSn

41.6.2.7 SPITXBUF Register (Offset = 10h) [Reset = 0000h]

SPITXBUF is shown in Figure 41-20 and described in Table 41-16.

Return to the Summary Table.

SPITXBUF stores the next character to be tranmitted. Writing to this register sets the TX BUF FULL Flag bit in SPISTS. When the transmission of the current character is complete, the contents of this register are automatically loaded in SPIDAT and the TX BUF FULL Flag is cleared. If no tranmission is currently active, data written to this register falls through into the SPIDAT register and the TX BUF FULL Flag is not set.
In CONTROLLER mode, if no tranmission is currently active, writing to this register initiates a transmission in the same manner that writing to SPIDAT does.

Figure 41-20. SPITXBUF Register

15	14	13	12	11	10	9	8
TXBn							
R/W-0h							
7	6	5	4	3	2	1	0
TXBn							
R/W-0h							

Table 41-16. SPITXBUF Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	TXBn	R/W	0h	Transmit Data Buffer This is where the next character to be transmitted is stored. When the transmission of the current character has completed, if the TX BUF FULL Flag bit is set, the contents of this register is automatically transferred to SPIDAT, and the TX BUF FULL Flag is cleared. Writes to SPITXBUF must be left-justified. Reset type: SYSRSn

41.6.2.8 SPIDAT Register (Offset = 12h) [Reset = 0000h]

SPIDAT is shown in Figure 41-21 and described in Table 41-17.

Return to the Summary Table.

SPIDAT is the transmit and receive shift register. Data written to SPIDAT is shifted out (MSB) on subsequent SPICLK cycles. For every bit (MSB) shifted out of the SPI, a bit is shifted into the LSB end of the shift register.

Figure 41-21. SPIDAT Register

15	14	13	12	11	10	9	8
SDATn							
R/W-0h							
7	6	5	4	3	2	1	0
SDATn							
R/W-0h							

Table 41-17. SPIDAT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	SDATn	R/W	0h	Serial Data Shift Register - It provides data to be output on the serial output pin if the TALK bit (SPICTL.1) is set. - When the SPI is operating as a CONTROLLER, a data transfer is initiated. When initiating a transfer, check the CLOCK POLARITY bit (SPICCR.6) described in Section 10.2.1.1 and the CLOCK PHASE bit (SPICTL.3) described in Section 10.2.1.2, for the requirements. In CONTROLLER mode, writing dummy data to SPIDAT initiates a receiver sequence. Since the data is not hardware-justified for characters shorter than sixteen bits, transmit data must be written in left-justified form, and received data read in right-justified form. Reset type: SYSRSn

41.6.2.9 SPIFFTX Register (Offset = 14h) [Reset = A000h]

SPIFFTX is shown in Figure 41-22 and described in Table 41-18.

Return to the Summary Table.

SPIFFTX contains both control and status bits related to the output FIFO buffer. This includes FIFO reset control, FIFO interrupt level control, FIFO level status, as well as FIFO interrupt enable and clear bits.

Figure 41-22. SPIFFTX Register

15	14	13	12	11	10	9	8
SPIRST	SPIFFENA	TXFIFO	TXFFST				
R/W-1h	R/W-0h	R/W-1h	R-0h				
7	6	5	4	3	2	1	0
TXFFINT	TXFFINTCLR	TXFFIENA	TXFFIL				
R-0h	W-0h	R/W-0h	R/W-0h				

Table 41-18. SPIFFTX Register Field Descriptions

Bit	Field	Type	Reset	Description
15	SPIRST	R/W	1h	SPI Reset Reset type: SYSRSn 0h (R/W) = Write 0 to reset the SPI transmit and receive channels. The SPI FIFO register configuration bits will be left as is. 1h (R/W) = SPI FIFO can resume transmit or receive. No effect to the SPI registers bits.
14	SPIFFENA	R/W	0h	SPI FIFO Enhancements Enable Reset type: SYSRSn 0h (R/W) = SPI FIFO enhancements are disabled. 1h (R/W) = SPI FIFO enhancements are enabled.
13	TXFIFO	R/W	1h	TX FIFO Reset Reset type: SYSRSn 0h (R/W) = Write 0 to reset the FIFO pointer to zero, and hold in reset. 1h (R/W) = Release transmit FIFO from reset.
12-8	TXFFST	R	0h	Transmit FIFO Status Reset type: SYSRSn 0h (R/W) = Transmit FIFO is empty. 1h (R/W) = Transmit FIFO has 1 word. 2h (R/W) = Transmit FIFO has 2 words. 10h (R/W) = Transmit FIFO has 16 words, which is the maximum. 1Fh (R/W) = Reserved.
7	TXFFINT	R	0h	TX FIFO Interrupt Flag Reset type: SYSRSn 0h (R/W) = TXFIFO interrupt has not occurred, This is a read-only bit. 1h (R/W) = TXFIFO interrupt has occurred, This is a read-only bit.
6	TXFFINTCLR	W	0h	TXFIFO Interrupt Clear Reset type: SYSRSn 0h (R/W) = Write 0 has no effect on TXFIFINT flag bit, Bit reads back a zero. 1h (R/W) = Write 1 to clear SPIFFTX[TXFFINT] flag.
5	TXFFIENA	R/W	0h	TX FIFO Interrupt Enable Reset type: SYSRSn 0h (R/W) = TX FIFO interrupt based on TXFFIL match (less than or equal to) will be disabled. 1h (R/W) = TX FIFO interrupt based on TXFFIL match (less than or equal to) will be enabled.

Table 41-18. SPIFFTX Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	TXFFIL	R/W	0h	Transmit FIFO Interrupt Level Bits Transmit FIFO will generate interrupt when the FIFO status bits (TXFFST4-0) and FIFO level bits (TXFFIL4-0) match (less than or equal to). Reset type: SYSRSn 0h (R/W) = A TX FIFO interrupt request is generated when there are no words remaining in the TX buffer. 1h (R/W) = A TX FIFO interrupt request is generated when there is 1 word or no words remaining in the TX buffer. 2h (R/W) = A TX FIFO interrupt request is generated when there is 2 words or fewer remaining in the TX buffer. 10h (R/W) = A TX FIFO interrupt request is generated when there are 16 words or fewer remaining in the TX buffer. 1Fh (R/W) = Reserved.

41.6.2.10 SPIFFRX Register (Offset = 16h) [Reset = 201Fh]

SPIFFRX is shown in Figure 41-23 and described in Table 41-19.

Return to the Summary Table.

SPIFFRX contains both control and status bits related to the input FIFO buffer. This includes FIFO reset control, FIFO interrupt level control, FIFO level status, as well as FIFO interrupt enable and clear bits.

Figure 41-23. SPIFFRX Register

15	14	13	12	11	10	9	8
RXFFOVF	RXFFOVFCLR	RXFIFORESET	RXFFST				
R-0h	W-0h	R/W-1h	R-0h				
7	6	5	4	3	2	1	0
RXFFINT	RXFFINTCLR	RXFFIENA	RXFFIL				
R-0h	W-0h	R/W-0h	R/W-1Fh				

Table 41-19. SPIFFRX Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RXFFOVF	R	0h	Receive FIFO Overflow Flag Reset type: SYSRSn 0h (R/W) = Receive FIFO has not overflowed. This is a read-only bit. 1h (R/W) = Receive FIFO has overflowed, read-only bit. More than 16 words have been received in to the FIFO, and the first received word is lost.
14	RXFFOVFCLR	W	0h	Receive FIFO Overflow Clear Reset type: SYSRSn 0h (R/W) = Write 0 does not affect RXFFOVF flag bit, Bit reads back a zero. 1h (R/W) = Write 1 to clear SPIFFRX[RXFFOVF].
13	RXFIFORESET	R/W	1h	Receive FIFO Reset Reset type: SYSRSn 0h (R/W) = Write 0 to reset the FIFO pointer to zero, and hold in reset. 1h (R/W) = Re-enable receive FIFO operation.
12-8	RXFFST	R	0h	Receive FIFO Status Reset type: SYSRSn 0h (R/W) = Receive FIFO is empty. 1h (R/W) = Receive FIFO has 1 word. 2h (R/W) = Receive FIFO has 2 words. 10h (R/W) = Receive FIFO has 16 words, which is the maximum. 1Fh (R/W) = Reserved.
7	RXFFINT	R	0h	Receive FIFO Interrupt Flag Reset type: SYSRSn 0h (R/W) = RXFIFO interrupt has not occurred. This is a read-only bit. 1h (R/W) = RXFIFO interrupt has occurred. This is a read-only bit.
6	RXFFINTCLR	W	0h	Receive FIFO Interrupt Clear Reset type: SYSRSn 0h (R/W) = Write 0 has no effect on RXFIFINT flag bit, Bit reads back a zero. 1h (R/W) = Write 1 to clear SPIFFRX[RXFFINT] flag
5	RXFFIENA	R/W	0h	RX FIFO Interrupt Enable Reset type: SYSRSn 0h (R/W) = RX FIFO interrupt based on RXFFIL match (greater than or equal to) will be disabled. 1h (R/W) = RX FIFO interrupt based on RXFFIL match (greater than or equal to) will be enabled.

Table 41-19. SPIFFRX Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	RXFFIL	R/W	1Fh	Receive FIFO Interrupt Level Bits Receive FIFO generates an interrupt when the FIFO status bits (RXFFST4-0) are greater than or equal to the FIFO level bits (RXFFIL4-0). The default value of these bits after reset is 11111. This avoids frequent interrupts after reset, as the receive FIFO will be empty most of the time. Reset type: SYSRSn 0h (R/W) = A RX FIFO interrupt request is generated when there is 0 or more words in the RX buffer. 1h (R/W) = A RX FIFO interrupt request is generated when there are 1 or more words in the RX buffer. 2h (R/W) = A RX FIFO interrupt request is generated when there are 2 or more words in the RX buffer. 10h (R/W) = A RX FIFO interrupt request is generated when there are 16 words in the RX buffer. 1Fh (R/W) = Reserved.

41.6.2.11 SPIFFCT Register (Offset = 18h) [Reset = 0000h]

SPIFFCT is shown in Figure 41-24 and described in Table 41-20.

Return to the Summary Table.

SPIFFCT controls the FIFO transmit delay bits.

Figure 41-24. SPIFFCT Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TXDLY							
R/W-0h							

Table 41-20. SPIFFCT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	TXDLY	R/W	0h	FIFO Transmit Delay Bits These bits define the delay between every transfer from FIFO transmit buffer to transmit shift register. The delay is defined in number SPI serial clock cycles. The 8-bit register could define a minimum delay of 0 serial clock cycles and a maximum of 255 serial clock cycles. In FIFO mode, the buffer (TXBUF) between the shift register and the FIFO should be filled only after the shift register has completed shifting of the last bit. This is required to pass on the delay between transfers to the data stream. In the FIFO mode TXBUF should not be treated as one additional level of buffer. Reset type: SYSRSn 0h (R/W) = The next word in the TX FIFO buffer is transferred to SPITXBUF immediately upon completion of transmission of the previous word. 1h (R/W) = The next word in the TX FIFO buffer is transferred to SPITXBUF1 serial clock cycle after completion of transmission of the previous word. 2h (R/W) = The next word in the TX FIFO buffer is transferred to SPITXBUF 2 serial clock cycles after completion of transmission of the previous word. FFh (R/W) = The next word in the TX FIFO buffer is transferred to SPITXBUF 255 serial clock cycles after completion of transmission of the previous word.

41.6.2.12 SPIPRI Register (Offset = 1Eh) [Reset = 0000h]

SPIPRI is shown in Figure 41-25 and described in Table 41-21.

Return to the Summary Table.

SPIPRI controls auxillary functions for the SPI including emulation control, SPIPTE inversion, and 3-wire control.

Figure 41-25. SPIPRI Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	SOFT	FREE	RESERVED		PTEINV	TRIWIRE
R-0h	R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h

Table 41-21. SPIPRI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	SOFT	R/W	0h	Emulation Soft Run This bit only has an effect when the FREE bit is 0. Reset type: SYSRSn 0h (R/W) = Transmission stops midway in the bit stream while TSUSPEND is asserted. Once TSUSPEND is deasserted without a system reset, the remainder of the bits pending in the DATBUF are shifted. Example: If SPIDAT has shifted 3 out of 8 bits, the communication freezes right there. However, if TSUSPEND is later deasserted without resetting the SPI, SPI starts transmitting from where it had stopped (fourth bit in this case) and will transmit 8 bits from that point. 1h (R/W) = If the emulation suspend occurs before the start of a transmission, (that is, before the first SPICLK pulse) then the transmission will not occur. If the emulation suspend occurs after the start of a transmission, then the data will be shifted out to completion. When the start of transmission occurs is dependent on the baud rate used. Standard SPI mode: Stop after transmitting the words in the shift register and buffer. That is, after TXBUF and SPIDAT are empty. In FIFO mode: Stop after transmitting the words in the shift register and buffer. That is, after TX FIFO and SPIDAT are empty.
4	FREE	R/W	0h	Emulation Free Run These bits determine what occurs when an emulation suspend occurs (for example, when the debugger hits a breakpoint). The peripheral can continue whatever it is doing (free-run mode) or, if in stop mode, it can either stop immediately or stop when the current operation (the current receive/transmit sequence) is complete. Reset type: SYSRSn 0h (R/W) = Emulation mode is selected by the SOFT bit 1h (R/W) = Free run, continue SPI operation regardless of suspend or when the suspend occurred.
3-2	RESERVED	R	0h	Reserved

Table 41-21. SPIPRI Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	PTEINV	R/W	0h	SPIPTEn Inversion Bit On devices with 2 or more SPI modules, inverting the SPIPTE signal on one of the modules allows the device to receive left and right-channel digital audio data. This bit is only applicable to PERIPHERAL mode. Writing to this bit while configured as CONTROLLER (CONTROLLER_PERIPHERAL = 1) has no effect Reset type: SYSRSn 0h (R/W) = SPIPTEn is active low (normal) 1h (R/W) = SPIPTE is active high (inverted)
0	TRIWIRE	R/W	0h	SPI 3-wire Mode Enable Reset type: SYSRSn 0h (R/W) = Normal 4-wire SPI mode. 1h (R/W) = 3-wire SPI mode enabled. The unused pin becomes a GPIO pin. In CONTROLLER mode, the SPIPICO pin becomes the SPICOCI (CONTROLLER receive and transmit) pin and SPIPOCI is free for non-SPI use. In PERIPHERAL mode, the SPIPOCI pin becomes the SPIPIPO (PERIPHERAL receive and transmit) pin and SPIPICO is free for non-SPI use.

Chapter 42

Single Edge Nibble Transmission (SENT)

This chapter describes the features and operation of the Single Edge Nibble Transmission Extended Receiver Features module (SENT). The SENT protocol is unidirectional and uses single wire between two or more points to transmit signals from one or more sensors to a controller. SENT uses the open standard released by the Society of Automotive Engineers (SAE) and supports J2716 January 2010 and J2716 April 2016, and is mainly used for automotive applications. This protocol can transmit high resolution data at a low cost to the system. The SENT module utilizes a Master Trigger Pulse Generator to control and receive data from one or more sensors, using a configurable pulse signal. The received data can be stored directly into memory or a FIFO and read by the CPU or RTDMA.

42.1 Introduction

The SENT module is based on the open standard SAE J2716 with additional enhancements such as additional sensor format support.

Note

The term 'channel' within this chapter and 'sensor' in the register descriptions are equivalent.

42.1.1 Features

The SENT module includes the following features:

- Based on SAE J2716 (J2716 January 2010 and J2716 April 2016)
- Supports 2007 and 2010 CRC checksum calculation
- Fast channel receiver
- Slow channel receiver
  - Short serial message (8-bit data and 4-bit message ID)
  - Enhanced serial 12-bit message (12-bit data and 8-bit message ID)
  - Enhanced serial 12-bit message (12-bit data and 8-bit message ID)
- Configurable memory depth
- Master Trigger Pulse Generator (MTPG) enables multiple sensors for the same SENT bus
- 5 SENT channels that can each be set to be triggered by one of 63 trigger sources
- Nibble sorting to minimize CPU intervention
- Timeout feature in SENT channel can be re-purposed for watchdog (only usable in continuous receive mode)
- RXD_I_R bit in the CSENT_RXD register is used for debugging 1 bit of the SENT receive at a time
- Time stamp captures for received data frames
  - Uses 32-bit free running counter
  - Can use external counter for one or all SENT modules
- Receiver and Interrupt Features
  - Programmable glitch filter on input (bypass mode available)
  - Automatic detection of CRC error and framing error on Fast and Slow Channel Data
  - Option to save data received with error
  - Configurable number of data nibbles to receive (1-8)
  - FIFO and direct map support for received data frames
  - RTDMA and interrupts can be used to send data depending on how full the FIFO is
  - Error Detection Supported:
    - Timeout
    - Calibration
    - FIFO Overflow/Underflow
    - Frequency Drift
    - Overflow Trigger Request

42.1.2 SENT Related Collateral

Foundational Materials

- C29x Academy - Single Edge Nibble Transmission (SENT)

42.2 Advanced Topologies: MTPG

The Master Trigger Pulse Generator (MTPG) generates pulses for synchronous mode sensors that sends data only after receiving a specific trigger pulse.

42.2.1 MTPG Features

- Supports variety of sensors which support synchronous access
- Supports up to four sensors on the same bus
- Broadcast Channel can configure the sensors (no response expected from sensor)
- Up to five sensor trigger channels (Broadcast, Channel 1-4), each of which has:
  - MTPG enable
  - Software trigger
  - Period value for indicating drive time for the specific waveform sequence
  - Timeout counter that triggers an interrupt when response is not received in the configured time limit
  - Nine compare registers for toggle specification and one receiver enable compare register
  - Trigger select
- 58 external triggers and 5 internal triggers
- Arbiter for selecting one of the five trigger pulse sources (round-robin priority)
- Wait time for MTPG to start after previous response

Note

The standard SENT configuration only uses a single channel (channel 1). Enabling the MTPG with the MTP_MODE bit in the RCFG2 register enables support for using multiple channels.

42.2.2 MTPG Description

The MTPG triggers can be used to configure sensors or request data from synchronous mode sensors, since the MTPG can create desired pulse patterns that are recognized by the sensors. For sensors with synchronous transmit, the sensor sends data when the MTPG has sent a pulse that matches with the sensor's pulse size requirement.

The MTPG enables the operational modes laid out in Table 42-1.

Table 42-1. MTPG Operational Modes

Mode	Maximum Number of Sensors Supported	Description
Synchronous Transmit	1	Sensor sends data only after receiving trigger pulse
Multiple Sensors on Bus	4	Each sensor has a specific address which the MTPG triggers in a predefined sequence. There can be a predefined broadcast sequence to configure the sensors
Change Mode of Operation	1	MTPG can change the mode of sensor operation

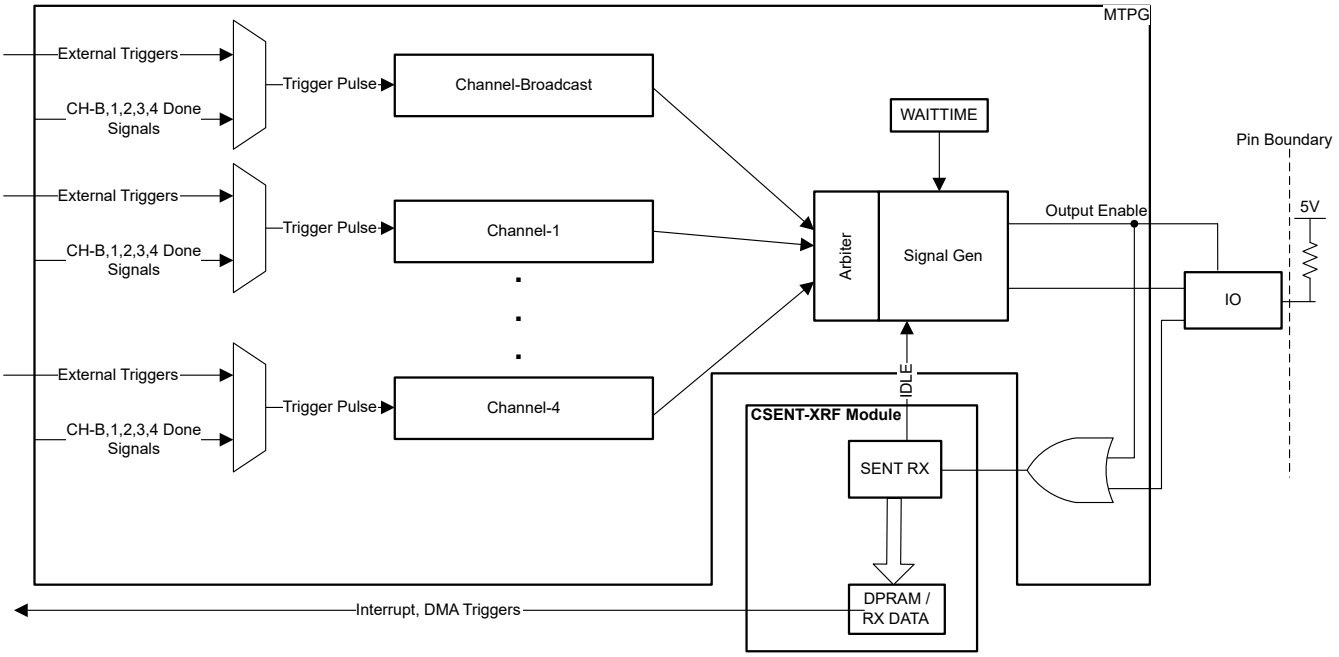


Figure 42-1. SENT Block Diagram

Figure 42-1 shows that the MTPG consists of one signal generator and five trigger channels (Broadcast and Channels 1-4). Each channel has a set of configurations that control the signal generation, and the signal generator uses one set of configurations to generate the pulse.

On the Broadcast channel the MTPG generates a pulse and the SENT receiver does not expect data in return. There can be a predefined broadcast of sequences to configure the sensors or to change the mode of the operation. In normal mode, the SENT receiver expects data in response to a pulse, and a timeout counter acts as a check to trigger an interrupt if there is no received sync pulse within a specified time limit. The SENT channel's period register defines the length of the pulse produced by the MTPG.

The channel has ten compare registers, which toggle the signal generator output when the channel counter matches any of the compare values. The maximum number of toggles that can be created is limited by the number of compare registers.

The CMP10 compare register is unique in that this compare register controls when the receiver is enabled to sample the RX line. The SENT peripheral can only drive the output when the period of the SENT channel's counter is non-zero.

Figure 42-2 shows the interaction between the output control, RX line, and the compare registers. To generate one trigger pulse, two compare registers must be non-zero. This is because the compare registers are triggered in sequence, so for example the CMP9 register must be larger than CMP8, which must be larger than CMP7, and so on. The purpose of this is to reduce the logic and length of the critical path. The respective Sn_MTP_CMP10RE register (where n is 1 through 4) is only used if the receiver input gating needs to occur earlier than the period time. The MTPG counter uses clock tick units.

Note

There is no BC_MTP_CMP10RE register since the broadcast channel does not expect a response.

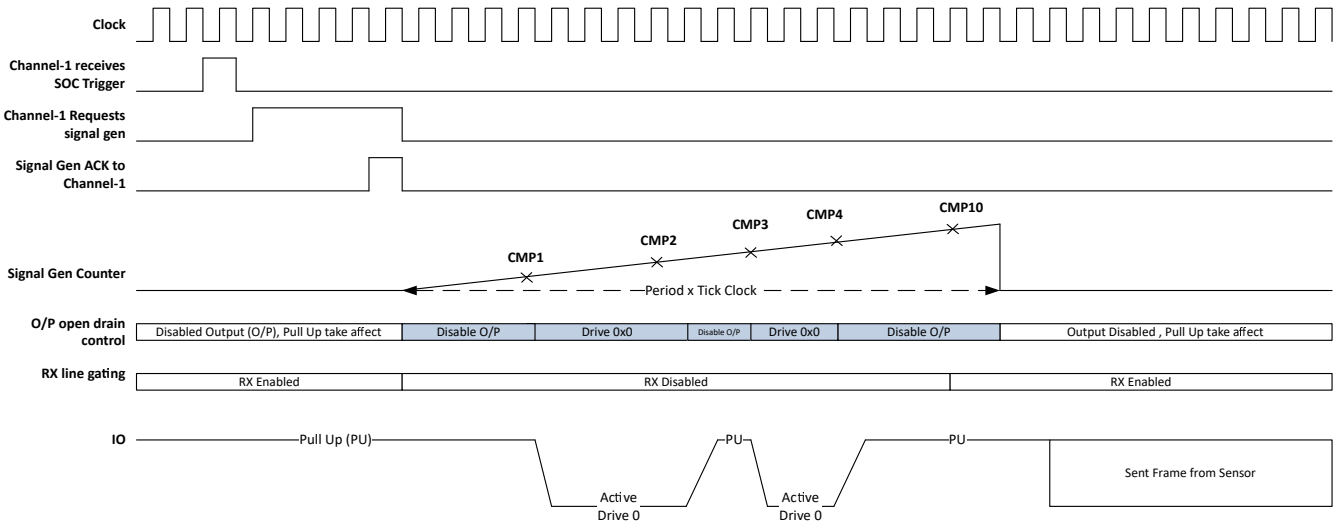


Figure 42-2. SENT Output Control

The signal generator initializes a counter to 0x0 after receiving a request from a channel to start the pulse sequence, and counts to the value stored in the channel's respective period register.

Note

The SENT pin must be configured for open-drain. This can be set using the GPIO open-drain register.

42.2.3 Channel Triggers

A channel is triggered by software, a trigger external to the SENT peripheral, or by another channel. See External Triggers for more details on the trigger sources for the SENT channels. The channel's done signal can be used to self-trigger, which allows continuous sampling of data. The use of another channel's done signal simplifies sequentially sampling sensors. When the SENT bus is idle and the trigger is received, the start of the pulse trigger is generated by the selected source from the TRIGSEL bitfield in the respective BC_TRIGSEL or Sn_TRIGSEL register (where n is 1 through 4). The trigger source from a channel is only valid when the channel's MTP_EN bit is set.

Note that because of the way the trigger connection is structured, all channels can be triggered by a single trigger, in which case the channels are serviced in a round-robin fashion. Valid trigger requests go through a round-robin to determine which channel trigger request is serviced, following this order of priority:

1. Broadcast Channel
2. Channel 1
3. Channel 2
4. Channel 3
5. Channel 4

After choosing the channel to be serviced, the valid trigger source for each channel can be checked using the TRIG_REQ bits in the TPGENSTAT register. If there is a trigger received before a pulse-sensor-response sequence completes, an overflow error occurs on the channel.

Once the arbiter chooses the channel to be serviced, the channel's configuration data is loaded and the trigger pulse is generated if the SENT peripheral is in idle or disabled. There is no wait time for the first trigger pulse serviced because there is no flag for the last received frame. After the first trigger pulse, the counter starts counting until reaching a configured wait time value in the WAITTIME register. This is a global register which controls the idle time between the last rising edge of the RX frame to the start of the MTPG pulse. This register can be a value from 0 to 511 and is in terms of SENT ticks.

Note

When SENT is disabled by RX_ENB, this indicates that the MTPG is used to enable the receiver after the pulse is sent. For this reason, a trigger pulse can be generated if the SENT peripheral is disabled.

The trigger changes the direction of the pin to be an output until the pulse sequence completes. The SENT receiver is gated when the pulse is generated so the pulse is not treated as a SENT frame.

Table 42-2. External Triggers

Index	Signal
0	Disable (no hardware triggers)
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	CPU1_TINT0
7	CPU1_TINT1
8	CPU1_TINT2
9	CPU2_TINT0
10	CPU2_TINT1
11	CPU2_TINT2
12	CPU3_TINT0
13	CPU3_TINT1
14	CPU3_TINT2
15-23	Reserved
24	INPUTXBAR5
25	INPUTXBAR11
26	INPUTXBAR12
27	INPUTXBAR13
28	INPUTXBAR14
29	INPUTXBAR15
30	INPUTXBAR16
31	EPWM1_ADCSOCA
32	EPWM1_ADCSOCB
33	EPWM2_ADCSOCA
34	EPWM2_ADCSOCB
35	EPWM3_ADCSOCA
36	EPWM3_ADCSOCB
37	EPWM4_ADCSOCA
38	EPWM4_ADCSOCB
39	EPWM5_ADCSOCA
40	EPWM5_ADCSOCB
41	EPWM6_ADCSOCA
42	EPWM6_ADCSOCB
43	EPWM7_ADCSOCA
44	EPWM7_ADCSOCB
45	EPWM8_ADCSOCA
46	EPWM8_ADCSOCB

Table 42-2. External Triggers (continued)

Index	Signal
47	EPWM9_ADCSOCA
48	EPWM9_ADCSOCB
49	EPWM10_ADCSOCA
50	EPWM10_ADCSOCB
51	EPWM11_ADCSOCA
52	EPWM11_ADCSOCB
53	EPWM12_ADCSOCA
54	EPWM12_ADCSOCB
55	EPWM13_ADCSOCA
56	EPWM13_ADCSOCB
57	EPWM14_ADCSOCA
58	EPWM14_ADCSOCB
59	EPWM15_ADCSOCA
60	EPWM15_ADCSOCB
61	EPWM16_ADCSOCA
62	EPWM16_ADCSOCB
63	EPWM17_ADCSOCA
64	EPWM17_ADCSOCB
65	EPWM18_ADCSOCA
66	EPWM18_ADCSOCB
67-94	Reserved
95	ECAP1_SOC
96	ECAP2_SOC
97	ECAP3_SOC
98	ECAP4_SOC
99	ECAP5_SOC
100	ECAP6_SOC
101-127	Reserved

42.2.4 Timeout

The timeout count down timer is an independent timer that can be used for the MTPG timeout and SENT timeout. The MTPG has the respective register Sn_MTP_TO where n is 1 through 4, for configuring the timeout value. The SENT peripheral has the CSENT_TO register for setting the timeout value.

Note

There is no BC_MTP_TO register since the broadcast channel does not expect a response.

For the MTPG, once the pulse is sent out, the timeout value of a non-broadcast channel is set. At the end of the period of the current trigger pulses, the timeout value is loaded to the counter to start counting down. If the RX line does not receive a low pulse within the timeout value, a timeout error occurs and creates an interrupt. This timeout can be configured for when there is no response from a sensor in a specified length of time, or on a per channel basis.

For the SENT peripheral, the timeout error occurs when SENT is idle for too long or the time between frames is longer than the timeout time. The value is set using the CSENT_TO register, and is loaded after the sensor frame is received.

Once the low pulse of the synchronization pulse is detected or the timeout expires, the countdown counter is disabled. The default timeout registers are set to 0, meaning the timeout countdown timer is disabled.

42.3 Protocol Description

SENT messages are encoded and decoded based on the time between falling or rising edges, depending on the idle polarity. For this section, the high idle polarity is described as an example. The same information applies for low idle polarity, given the edge direction of the clock is swapped. The messages are timed using clock tick time, or TT, which is calculated during the calibration/synchronization pulse.

A slow frame is composed of multiple fast frames, and transmits longer streams of data. The slow frame also utilizes multiple fast frames to transmit additional information through the status and communication nibble, such as the message ID, slow data, and CRC.

The frame format for SENT is as follows:

1. Calibration or synchronization pulse with a fixed period (56 clock ticks)
   a. This is used by the receiver as a reference to measure the TT. The pulse starts with a falling edge for high idle polarity and rising edge for low idle polarity and the counter for the synchronization period starts counting. For a high idle polarity, the period between the first falling edge and the second falling edge is divided into 56 to determine the TT. The TT is used to sample the nibbles' values. If more than one synchronization pulse is detected, the first one is ignored under the assumption the pulse is a pause pulse.
2. One status and serial communication nibble pulse (12 to 27 clock ticks)
   a. The status nibble transmits miscellaneous information such as part numbers, mode of operation, and error code information in slow channel mode (1-bit per frame serial message). This nibble is not included for the frame's CRC calculation if the RX_CRC_WITH_STATUS bit is not set.

Table 42-3. Status Nibble Description

Bit	Functionality
0	Channel 2 error indicator (1 = error)
1	Channel 1 error indicator (1 = error)
2	Serial data message bit (CRC, message ID, and data)
3	Short Serial Message Pattern: Start with 1 and the rest of the 15 messages are 0 Enhanced Serial Message Pattern: Unique pattern [0]1111110, message 13 and 18 are 0; the [0] is a zero pre-condition before the start of serial message 1

   i. The status nibble for the FAST channel without a slow serial message is 0, with an option for using bit 0 and/or bit 1 as an error flag for certain sensors. If the sensor is in error, bit 0 or 1 of the status nibble is set to 1.
   ii. A Short or Enhanced Serial Message is ignored on the status and communication nibble until the appropriate pattern of the message is received. Note that if the status nibble is used in a way not defined by the J2716 standard, then mask the serial errors appropriately.
3. Sequence of one to eight data nibble pulses (12 to 27 clock ticks for each nibble pulse)
   a. For high idle polarity, the pulse starts with a falling edge and remains low for a few clock ticks. Then each nibble's value is determined between falling edges, with the minimum pulse period being 12 clock ticks and the maximum being 27 clock ticks. The incoming serial nibble value is sampled at a clock tick length of TT / 2. The sampling occurs at the middle of the nibble pulse. For high idle polarity, the SENT TT counter counts the number of TT between two falling edges of the nibble pause and offsets the value with 12.

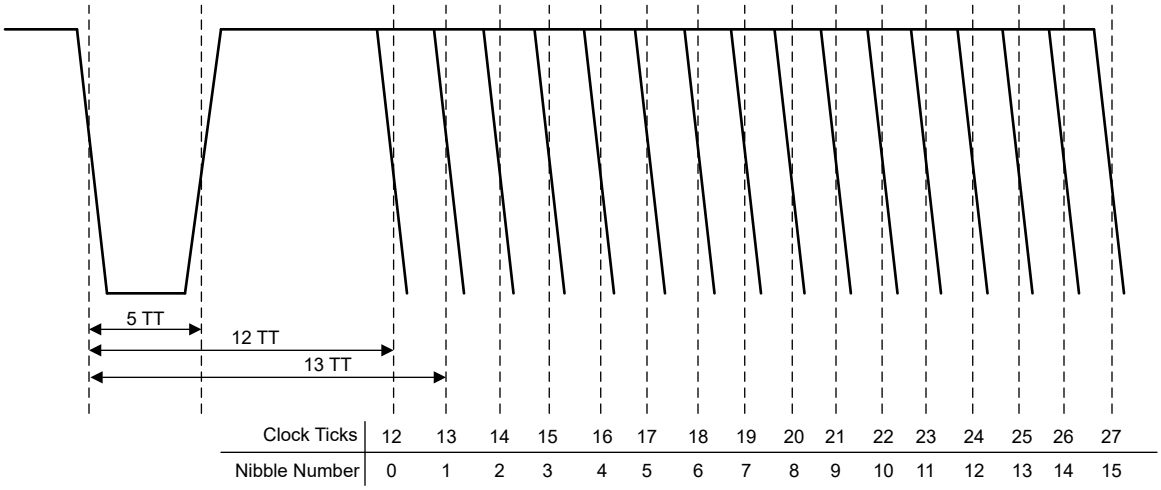


Figure 42-3. Nibble Pulse

   b. If the number of TT between the two falling edges is less than 11 or greater than 27, the value is invalid and the frame error is set.
4. One CRC nibble pulse (12 to 27 clock ticks)
   a. See Section 42.3.2.
5. Pause pulse (optional)
   a. This is an optional part of the SENT message format. For a fixed-length frame, the pause pulse fills up the message after the checksum nibble. This is the portion between 12 clock ticks and 768 clock ticks. If the RX_PPENB is enabled in the RCFG register the receiver expects the next pulse to be a pause pulse after the CRC nibble. When no pause pulse is enabled, the receiver expects that a packet received is immediately followed by another packet calibration/synchronization pulse. This does not apply for MTP_MODE, where an end pulse is expected at the end of the final CRC field.
6. End pulse for synchronous mode SENT that requires MTPG (12 ticks)
   a. For sensors synchronously responding to the MTPG, a 12-clock tick end pulse terminates the frame transmission after the CRC nibble and idly waits for the next trigger pulse. The SENT peripheral does not require a complete end pulse, and acknowledges the received packet when the CRC is verified.

42.3.1 Nibble Frame Format

A sensor's frame can contain one or two sensor data channels. The SENT receives data nibbles according to the format sent by the connected sensor. The DATA0_MAP and DATA1_MAP registers configure the nibble sorting and determine whether the received nibble is part of data 0 or data 1 memory map, as well as where the data nibble, status nibble, or CRC nibble are stored in the corresponding FIFO or direct memory location in SENT_MEM.

Figure 42-4 shows the format for a fast SENT message. For the Short Serial Message Format or Enhanced Serial Message Format, this is a single frame of the entire message.

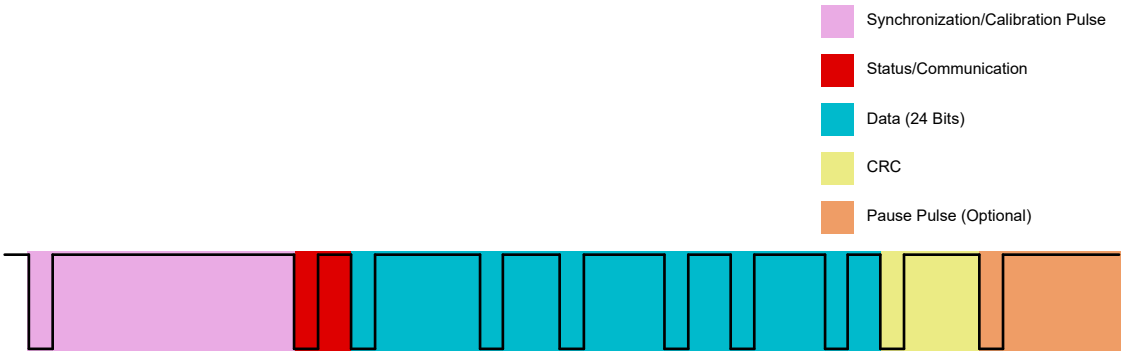


Figure 42-4. Fast Message

Table 42-4. SENT Data Frame Format Examples

Frame Format	Data Nibbles	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6
Two 12-bit fast channels	6	Ch1 Most-Significant Nibble	Ch1 Middle-Significant Nibble	Ch1 Least-Significant Nibble	Ch2 Most-Significant Nibble	Ch2 Middle-Significant Nibble	Ch2 Least-Significant Nibble
One 12-bit fast channel	3	Ch1 Most-Significant Nibble	Ch1 Middle-Significant Nibble	Ch1 Least-Significant Nibble	N/A	N/A	N/A
High-speed with one 12-bit fast channel	4	Bits [11..9]	Bits [8..6]	Bits [5..3]	Bits [2..0]	N/A	N/A
Secure sensor with 12-bit fast channel 1 and secure sensor information on fast channel 2	6	Ch1 Most-Significant Nibble	Ch1 Middle-Significant Nibble	Ch1 Least-Significant Nibble	Counter Most-Significant Nibble	Counter Least-Significant Nibble	Inverted Copy of Ch1 Most-Significant Nibble
Single sensor with 12-bit fast channel 1 and no value on fast channel 2	6	Ch1 Most-Significant Nibble	Ch1 Middle-Significant Nibble	Ch1 Least-Significant Nibble	0	0	0
14-bit fast channel 1 and 10-bit fast channel 2	6	Ch1 Most-Significant Nibble	Ch1 Middle Most-Significant Nibble	Ch1 Middle Least-Significant Nibble	Ch1/Ch2 Least-Significant Nibble	Ch2 Middle-Significant Nibble	Ch2 Least-Significant Nibble
16-bit fast channel 1 and 8-bit fast channel 2	6	Ch1 Most-Significant Nibble	Ch1 Middle Most-Significant Nibble	Ch1 Middle Least-Significant Nibble	Ch2 Least-Significant Nibble	Ch2 Least-Significant Nibble	Ch2 Most-Significant Nibble

An example of the data mapping for two 12-bit fast channels frame format is shown in Table 42-5. This table uses an address offset, which is relevant to the corresponding base address for the SENT peripheral used.

Table 42-5. Data Mapping for Two 12-Bit Fast Channels

Address Offset	Byte 3	Byte 2	Byte 1	Byte 0	Word
0x0	Timestamp[31..24]	Timestamp[23..16]	Timestamp[15..8]	Timestamp[7..0]	Timestamp
0x4	0x00	0x00	0x0, Data Nibble 1	Data Nibble 2, Data Nibble 3	Data 0
0x8	0x00	0x00	0x0, Data Nibble 6	Data Nibble 5, Data Nibble 4	Data 1

42.3.2 CRC

The Cyclic Redundancy Check, or CRC, is a method of detecting errors in transmitted data. The CRC checksum is used when the RX_CRCENB bit in the RCFG register is enabled. CRC4, CRC6, and CRC8 are used for the receiver to check against the received data. The corresponding CRC check outputs a 4-bit, 6-bit, or 8-bit CRC respectively.

1. CRC4
   a. The CRC4 value is used to verify the validity of data nibbles in the Fast Channel frame, the message ID, and the 8-bit data in the Slow Channel Short Serial Message. CRC4 uses polynomial G(x) = x^4 + x^3 + x^2 + 1 with a seed value of 0101. The status and communication nibbles are included in this CRC calculation if the RX_CRC_WITH_STATUS parameter or programmable bit is enabled. The checksum is implemented as a bit-wise XOR with a 16-element array lookup. The checksum is determined by using all data nibbles in sequence and check-summing the result with an extra zero value. A 4-bit zero nibble is added after the data nibbles to protect for common errors in the last data nibble and checksum.
2. CRC6
   a. The CRC6 value is used to verify the validity of the 24-bit Slow Channel Enhanced Serial Message using the polynomial G(x) = x^6 + x^4 + x^3 + 1 with a seed value of 010101. CRC6 is also used to verify the 24-bit Fast Channel Frame format data using a different polynomial, G(x) = x^6 + x + 1 with a seed value of 010101. The checksum is implemented using a bit-wise XOR with a 64-array lookup. The checksum is determined by reading in 6-bit groups of the 24-bit message data in order and check-summing the result with an extra zero value, as mentioned in Section 42.3.5. The 6-bit zero nibble is added after the data nibbles to protect for common errors in the last data nibble and checksum.
3. CRC8
   a. The CRC8 value is used to verify the validity of the 28-bit Fast Channel Frame format data, which contains 7 data nibbles and 2x2 Rolling counter data. This checksum uses the polynomial G(x) = x^8 + x^5 + x^3 + x^2 + x + 1 with a seed value of 01010101.

42.3.3 Short Serial Message Format

The short serial message format uses multiple fast SENT data frames to transmit slow data. While the fast data is transmitted through each single SENT frame, the slow data is transmitted using bit 2 of the status and communication nibble pulse, with the most-significant bit first. The serial data is communicated in a 16-bit sequence. This means that after 16 SENT frames are received correctly, there is a 4-bit message ID, an 8-bit slow data, and a 4-bit CRC. The start bit of the serial message is indicated by a 1 in bit 3 of the status and communication nibble pulse, and the subsequent 15 frames must have a status bit 3 which is 0.

Note

To receive serial data correctly, all 16 frames must have no errors relating to the calibration pulse, CRC, frame, or other error types. This is verified using the RSLOW_DV flag.

The 16-bit message contains a 4-bit message ID nibble, 2 nibbles of data, and a CRC nibble. The CRC checksum is derived from the message ID and 2 data nibbles.

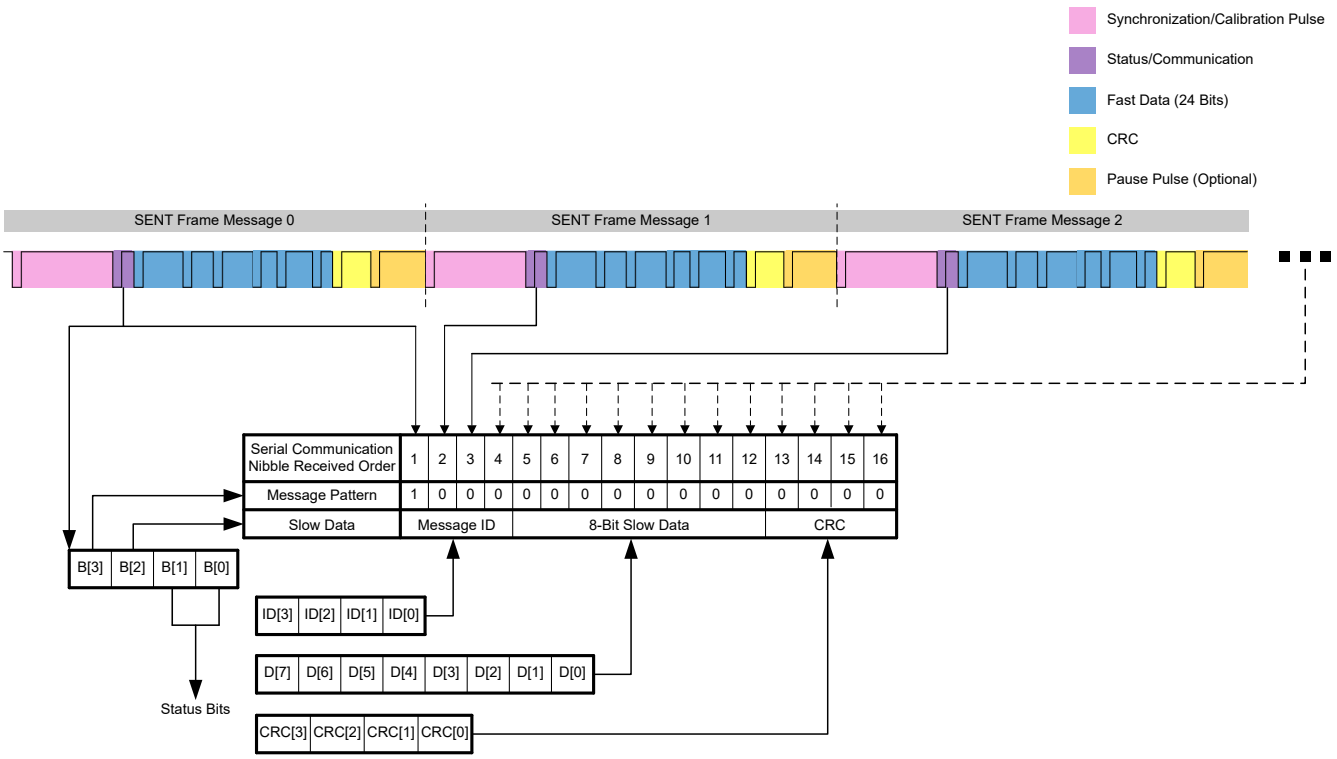


Figure 42-5. Short Serial Message

The message ID indicates the type of data in the data byte, which is ordered most-significant bit first.

Table 42-6. Message IDs

Message ID	Sensed Value for Data Field
0000	High Byte of Air Temperature
0001	Low Byte of Air Temperature
0010	High Byte of Humidity
0011	Low Byte of Humidity
0100	High Byte of Barometric Pressure Data
0101	Low Byte of Barometric Pressure Data
0110-1110	Reserved
1111	Error Codes

42.3.4 Enhanced Serial Message Format

The Enhanced Serial Message Format has a larger data field and set of message IDs available than the Short Serial Message Format. The serial message frame consists of a configuration bit and configurable 20-bit payload. The Enhanced Serial Message Format allows for this 20-bit payload to be configurable as an 8-bit message ID with 12-bit slow data or a 4-bit message ID with 16-bit slow data. The serial data is transmitted in bit 2 and 3 of the status and communication nibble from 18 consecutive data frames, similar to the Short Serial Message Format. The serial message is interleaved in 18 frames, so SENT keeps track and extracts the status and communication bits from every frame. Thus, by the 18th frame the complete message is received, given there are no errors.

Note

To receive serial data correctly, all 18 frames must have no errors, such as calibration error, CRC error, frame error, format error, or a zero before the start. The slow message counter and the FSM is reset when the message does not have zero before the start, "111110" after the start, or message 13 or 18 does not have 0 in bit 3 of the status and communication nibble.

The start of a serial message frame is indicated by "01111110" in bit 3 of the status and communication nibble. The first 1 in the sequence indicates the first nibble in the serial message frame. Bit 3 of the status and communication nibble must be 0 for message 7, 13, and 18. Bit 3 of the status and communication nibble of frame 8 is the C bit, which chooses the configuration.

1. C = 0 indicates 12-bit data and 8-bit message ID
2. C = 1 indicates 16-bit data and 4-bit message ID

The advantage to using the Enhanced Serial Message Format is that either a larger selection of message IDs can be used or more slow data can be transmitted using bit 3 of the status and communication nibble pulse, but in both cases there is more slow data received than the Short Serial Message Format.

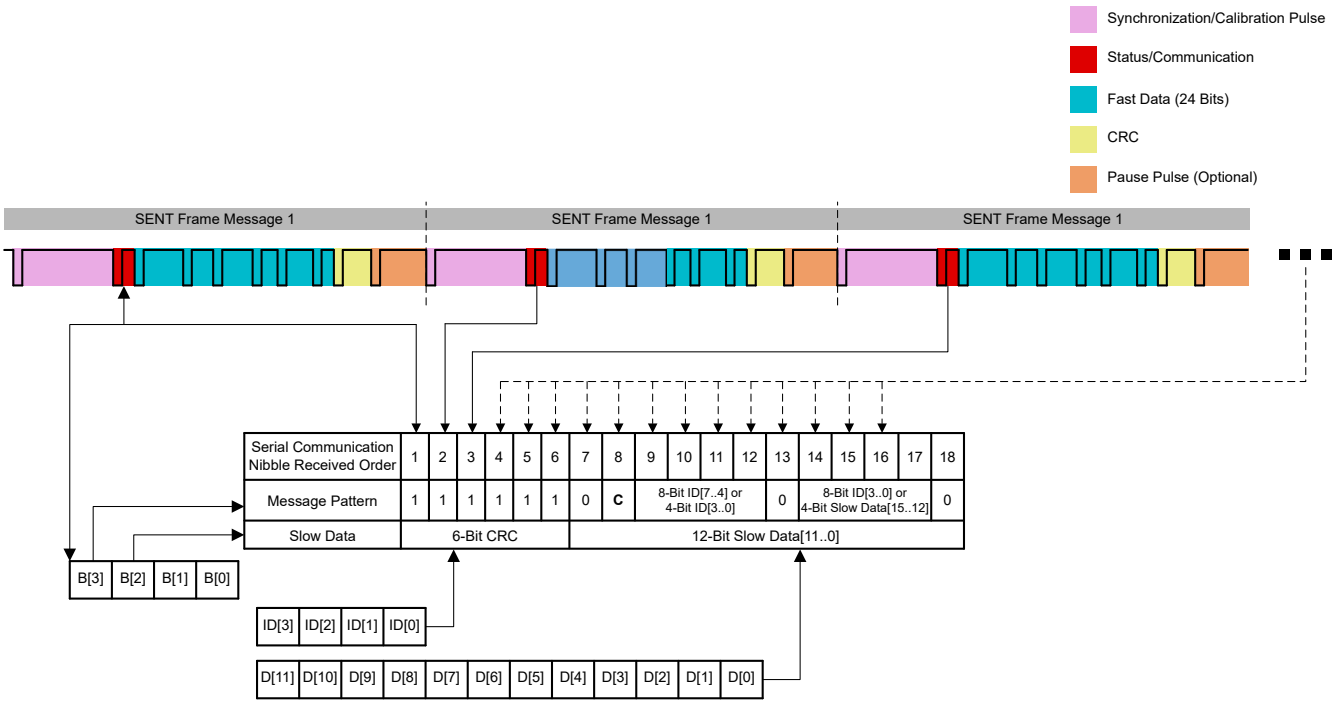


Figure 42-6. Enhanced Serial Message

42.3.5 Enhanced Serial Message Format CRC

The 6-bit CRC value is the result of the CRC calculation of the serial data message bit 2 and bit 3 in message frame 7 through 18. The polynomial G(x) = x^6 + x^4 + x^3 + 1 with a seed value 010101 is used. Six zeros are added to the message for CRC generation: Mcrc = [m0, m1, m2, … m23, 0, 0, 0, 0, 0, 0].

Serial Communication Nibble Received Order	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Start Bit (bit 3)	1	1	1	1	1	1	0	C	ID7	ID6	ID5	ID4	0	ID3	ID2	ID1	ID0	0
Serial Data (bit 2)	6-Bit CRC						D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

24-Bit Data for CRC Generation

Message CRC Bit	m0	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13	m14	m15	m16	m17	m18	m19	m20	m21	m22	m23
Value	D11	0	D10	C	D9	ID7	D8	ID6	D7	ID5	D6	ID4	D5	0	D4	ID3	D3	ID2	D2	ID1	D1	ID0	D0	0

First Data Bit

Figure 42-7. Enhanced Serial Message Data Order for CRC Generation

42.3.6 Receive Modes

The SENT peripheral decodes serial data and performs a CRC check before the data is stored in the receive buffer. The decoding includes arranging the module before writing the data to the buffer, such as when a Fast Channel frame is received. The SENT frame format varies among different types of sensors, with the greatest number of nibbles being 11 (two CRC nibbles, eight data nibbles, and one status nibble).

The receive buffer for the SENT peripheral can be configured as a FIFO or a direct map. The received data is stored in the SENT_MEM location, with the offset depending on the receive frame and the mode chosen.

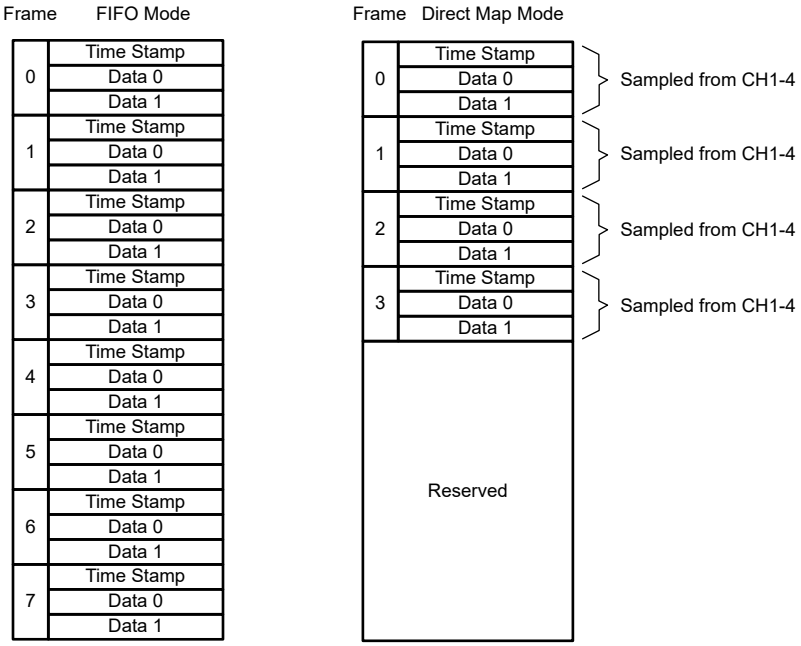


Figure 42-8. FIFO vs Direct Map Receive Modes

The FIFO mode is enabled by the FIFO_MODE bit in the RCFG2 register. In FIFO mode, the FIFO supports a depth of 24 values. This mode allows the host to be slow without causing data loss. Each sensor can store up to two data per frame, in addition to the time stamp. This allows 8 individual frames of data to be store using FIFO mode. The FIFO level is an indication of the number of 32-bit words available in the buffer, where each frame consists of three 32-bit words (timestamp, data 0, and data 1). This FIFO level can be used to generate an interrupt, if the level meets or exceeds the user configured threshold in RFIFO_TRIGLEV. See Section 42.5 for more details.

If the receiver FIFO receives a valid frame while the FIFO is full, the RFIFO_OVFERR bit in the RINTFLAG register is set and causes a receiver error interrupt. If the CPU or RTDMA try to read the FIFO when the FIFO is empty, the RFIFO_UNDFERR bit in the RINTFLAG register is set and causes a receiver error interrupt.

If the FIFO mode is disabled, the received data is directly mapped to a buffer at a specific address (see Section 42.8 for more details). In the direct map configuration, target devices can be triggered in any order. Upon a receive interrupt, the data is grabbed by the CPU from the specific memory location.

42.4 RTDMA Trigger

The SENT RTDMA trigger creates a transfer request to the RTDMA. The DMA_ENB bit must be enabled and the FIFO level must be greater than or equal to the user-configured threshold in RFIFO_TRIGLEV to trigger this RTDMA request. Disabling the RTDMA trigger resets the trigger logic. Figure 42-9 shows an example of implementing a RTDMA trigger for the SENT peripheral.

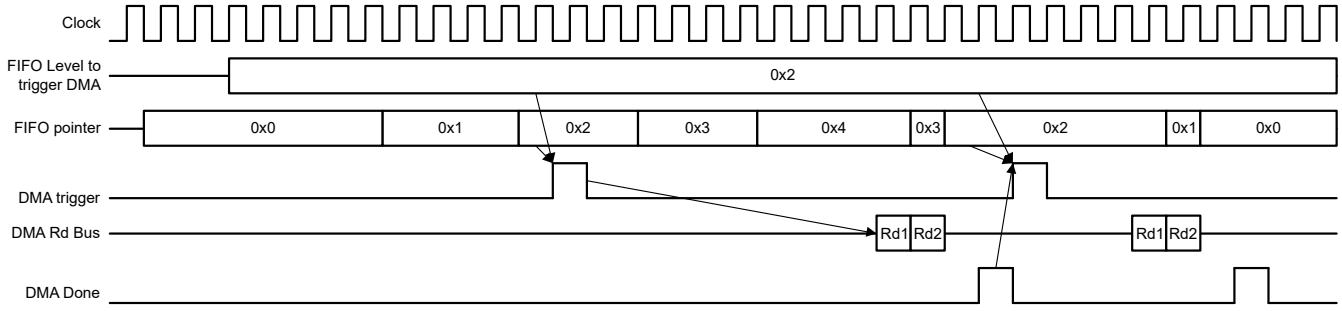


Figure 42-9. RTDMA Trigger Example

42.5 Interrupts Configuration

The SENT peripheral has interrupts for receive errors and FIFO thresholds. The RINTFLAG register is updated when there are errors in the received data, and the RSTAT register is updated with the relevant status information is made available.

An interrupt can be created based on the FIFO level of the SENT peripheral when the interrupt is enabled and the FIFO level is greater than or equal to the user-configured threshold in RFIFO_TRIGLEV. Figure 42-10 shows an example of implementing an interrupt trigger for the SENT peripheral with a FIFO threshold.

When a receive error occurs the entire packet or frame is not considered valid, depending on the NOWR_ERRDATA in the RCFG2 register. Frame data with errors are still saved to the receive buffer if NOWR_ERRDATA is 0. Receive error flags can occur simultaneously, with the RINTFLAG register showing the cause of the flags raised. The following errors activate the error interrupt flags:

- Fast CRC Error: The receiver CRC calculation does not match the CRC nibble received of the fast channel; enabled by RX_CRCENB in the RCFG register
- Fast Frame Error: The received status or data nibble of the fast channel is less than 12 TT or greater than 27 TT
- Slow CRC Error: The received slow channel message ID and data CRC calculation does not match the received slow channel CRC; enabled by RX_CRCENB in the RCFG register
- Slow Format Error: The received slow channel message contains an incorrect format than what is expected. In this case, the state machine for detecting the correct sequence for a slow frame restarts each time there is an error (this way the state machine gets properly synchronized again with the calibration pulse after enough restarts)

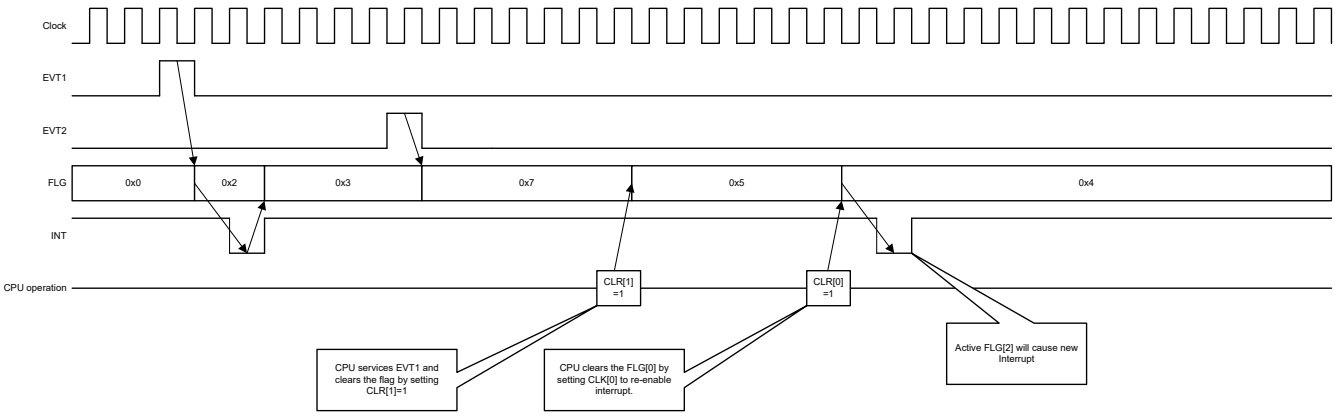


Figure 42-10. Interrupt Trigger Example

The following list shows additional conditions which, when detected, can generate an error or interrupt:

1. Calibration errors (two types)
   a. Type 1: Receiver is programmed with initial tick-time clock cycle according to the TTCLK bits in the RCFG register, which is derived from the calibration pulse of a frame. If the newly determined calibration TT differs by more than 25% from the initially programmed time, the RFAST_SYNCERR25 flag is raised. The receiver assumes the newly derived TT going forward unless the TTCLK is re-programmed.
   b. Type 2: Synchronization pulse conversion with the previous frame differs by more than 1.5625%, then the RFAST_SYNCERR flag is raised.
2. Too many/few nibbles between calibration pulses
   a. A RFAST_SnFRME error can occur (where n can be 1 through 4) when there are too few nibbles between calibration pulses, or when the received nibble is <12 ticks or >27 ticks; a RFAST_SnCRCE error can occur if there are too many nibbles and the RX_CRCENB is enabled. This can happen because the next nibble beyond the expected and configured value gets interpreted as a CRC. There is no frame error because the receiver is expecting a valid calibration pulse after the extra nibble
3. Frequency drift
   a. If two successive frames vary in length by more than 1.5625%, the FREQDRIFT_ERR flag is raised. This is determined by a ratio of the calibration pulses to the message length. This condition occurs if this ratio varies by more than $\frac{1}{64}$ or less than $-\frac{1}{64}$. This interrupt must be masked unless you are confident there are fixed length frames, which can be created by included a pause pulse. This check is active if the pause pulse RX_PPENB is expected and enabled.

The interrupt architecture for the SENT peripheral is shown in Figure 42-11.

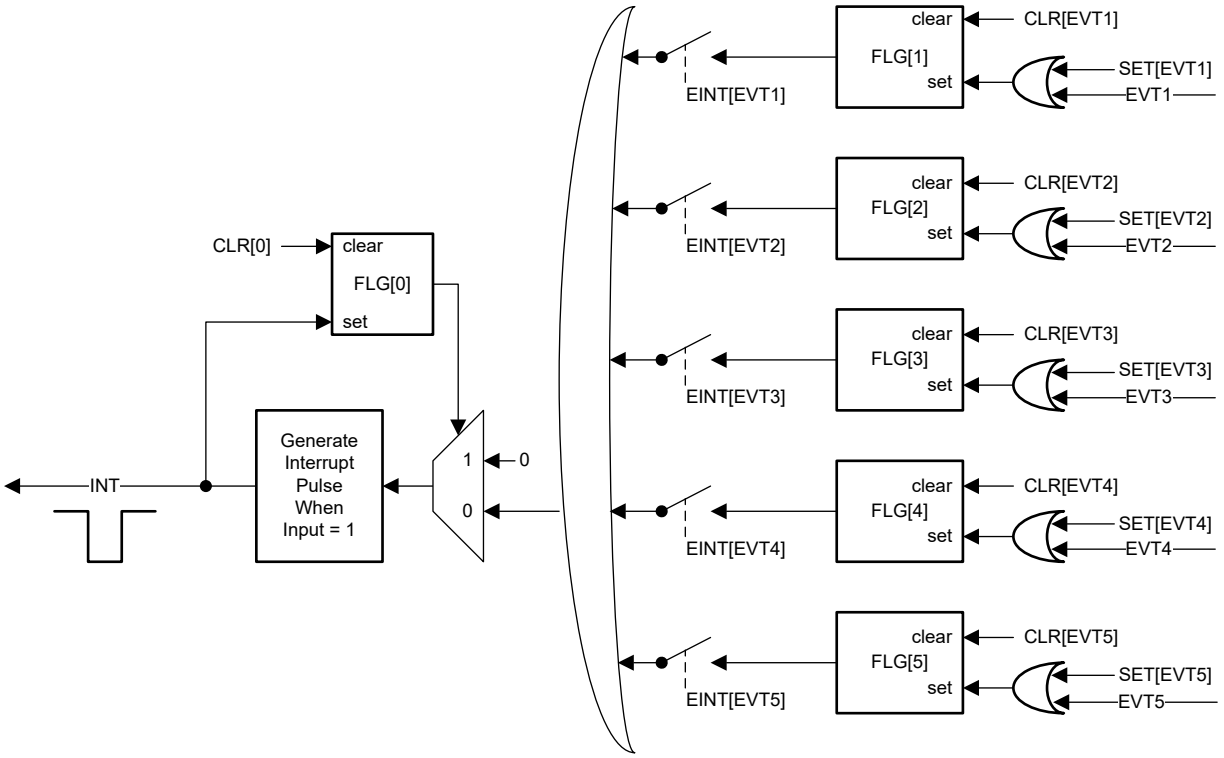


Figure 42-11. Interrupt Block Diagram

Table 42-7. Interrupt Types

Type	Flag	Event[1] [2]	Condition	Clear Interrupt
Valid Receives	GLBL	GLBL_flag0	This flag generates a low pulse interrupt flag	RCLRINT[0]
	RFAST_S1DV	rs1valid_evt	Active when SENT is in multi sensor mode and receives the corresponding channel's fast channel frame or standard SENT[3] frame data without any errors	RCLRINT[1]
	RFAST_S2DV	rs2valid_evt		RCLRINT[2]
	RFAST_S3DV	rs3valid_evt		RCLRINT[3]
	RFAST_S4DV	rs4valid_evt		RCLRINT[4]
	RSLOW_DV	rslow_valid_evt	Set when SENT receives slow channel data (16 or 18 consecutive fast channel data) without any errors	RCLRINT[5]

Table 42-7. Interrupt Types (continued)

Type	Flag	Event[1] [2]	Condition	Clear Interrupt
Receive Errors	RSLOW_CRCERR	rslow_crcerr_evt	The RSLOW_CRCERR is set when the received slow channel message ID and data CRC calculation does not match the received slow channel CRC; RX_CRCENB in the RCFG register must be enabled	RCLRINT[6]
	RSLOW_FORMATERR	rslow_formaterr_evt	The RSLOW_FORMATERR is set when the received Slow Channel message contains wrong format	RCLRINT[7]
	RFAST_S1CRCE	rfast_crcerr_evt	The RFAST_CRCERR is set when the Receiver CRC calculation (RX_CRCENB must be enabled) does not match the CRC nibble received on the corresponding channel or standard SENT[3] fast channel	RCLRINT[8]
	RFAST_S2CRCE	rfast_crcerr_evt		RCLRINT[9]
	RFAST_S3CRCE	rfast_crcerr_evt		RCLRINT[10]
	RFAST_S4CRCE	rfast_crcerr_evt		RCLRINT[11]
	RFAST_S1FRME	rfast_frmerr_evt	The RFAST_FRMERR is set when the received status or data nibble of the Sensor1 or standard SENT[3] fast channel is less than 12 TT or greater than 27 TT, which is invalid	RCLRINT[12]
	RFAST_S2FRME	rfast_frmerr_evt		RCLRINT[13]
	RFAST_S3FRME	rfast_frmerr_evt		RCLRINT[14]
	RFAST_S4FRME	rfast_frmerr_evt		RCLRINT[15]
	RTIMEOUT_ERR[0]	rtimeout_err_evt[0]	The RTIMEOUT_ERR[0] is set when there is no response, since the MTPG is sent out to the channel, or there is an unexpectedly long idle time between frames for standard SENT mode[3]	RCLRINT[16]
	RTIMEOUT_ERR[1]	rtimeout_err_evt[1]		RCLRINT[17]
	RTIMEOUT_ERR[2]	rtimeout_err_evt[2]		RCLRINT[18]
	RTIMEOUT_ERR[3]	rtimeout_err_evt[3]		RCLRINT[19]
	RFAST_SYNCERR25	rfast_syncerr25_evt	The RFAST_SYNCERR is set when the received Calibration or Synchronization pulse is more than +/-25% from the nominal 56 ticks	RCLRINT[20]
	RFAST_SYNCERR	rfast_syncerr_evt	The RFAST_SYNCERR is set when the received Calibration or Synchronization pulse is more than +/-25% from the nominal 56 ticks or the successive calibration pulse differ by more than 1.5625%	RCLRINT[21]
	FREQDRIFT_ERR	freqdrift_err_evt	The FREQDRIFT_ERR is set when the frequency drift on the fixed length frames (the ratio of the calibration pulse to the message length varies by more than 1/64 or less than -1/64 from one message to another)	RCLRINT[22]
	RFIFO_TRIGGER	rfifo_trig_evt	The RFIFO_TRIGGER is set when the Receiver FIFO's pointer is greater than or equal to the programmed trigger level	RCLRINT[23]
	RFIFO_OVFERR	rfifo_ovferr_evt	The RFIFO_OVFERR is set when the Receiver FIFO is overflow	RCLRINT[24]
	RFIFO_UNDFERR	rfifo_undferr_evt	The RFIFO_UNDFERR is set when the Receiver FIFO is underflow	RCLRINT[25]
MTPG Errors	OVFTRIG_ERR[0]	ovftrig_err_evt[0]	The OVFTRIG_ERR is set when the Broadcast Channel of the MTPG has Overflow trigger requests	RCLRINT[26]
	OVFTRIG_ERR[1]	ovftrig_err_evt[1]	The OVFTRIG_ERR is set when the corresponding channel of the MTPG has Overflow trigger requests	RCLRINT[27]
	OVFTRIG_ERR[2]	ovftrig_err_evt[2]		RCLRINT[28]
	OVFTRIG_ERR[3]	ovftrig_err_evt[3]		RCLRINT[29]
	OVFTRIG_ERR[4]	ovftrig_err_evt[4]		RCLRINT[30]

(1) The appropriate REINT bit must be set for the event to occur
(2) The RSETINT bit corresponding to the RINTFLAG bit can be set to trigger the event/flag manually, and the same applies to the RCLRINT to clear the event/flag manually

(3) Standard SENT mode in the table refers to using SENT without the MTPG to trigger a pulse

42.6 Glitch Filter

The GFILT field in the RCFG2 register contains the number of SYSCLKS that the received serial data input is filtered by. The glitch filter can cause a delay up to GFILT SYSCLKS, but this is usually negligible in terms of TT resolution.

In Figure 42-12, the GFILT is programmed to be 5. For every change on the receive input that is less than or equal to this value, the change is filtered out. The number of clock cycles per tick period is normally much larger than GFILT.

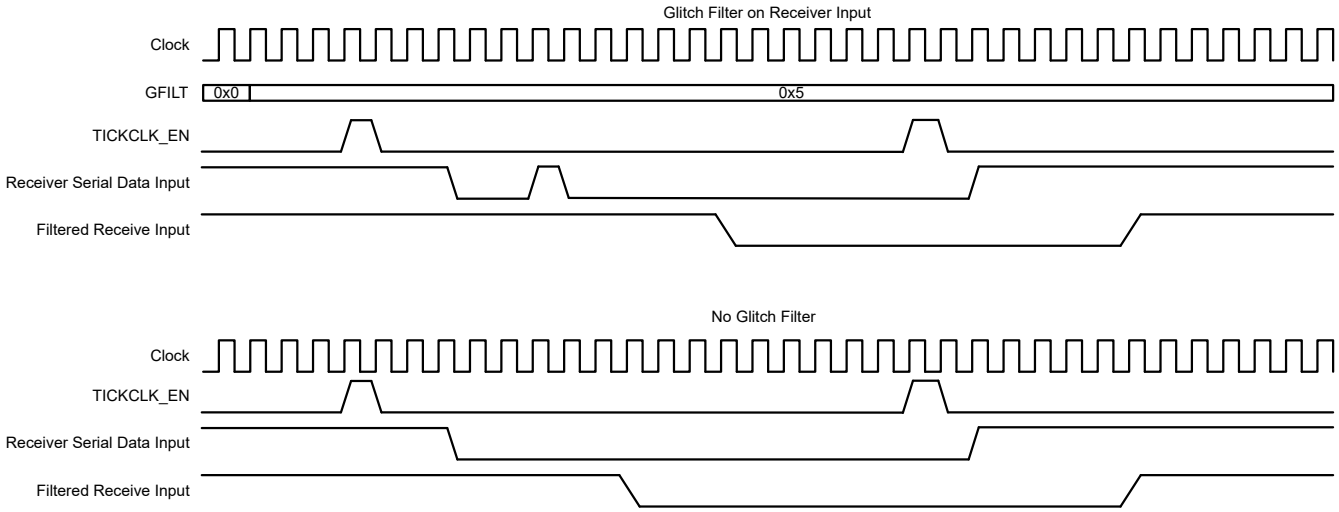


Figure 42-12. Glitch Filter Examples

42.7 Software

42.7.1 SENT Registers to Driverlib Functions

Table 42-8. SENT Registers to Driverlib Functions

File	Driverlib Function
RCFG	
sent.h	SENT_enableSENTReceiver
sent.h	SENT_disableSENTReceiver
sent.h	SENT_enableSENTCRC
sent.h	SENT_disableSENTCRC
sent.h	SENT_enableSENTPausePulse
sent.h	SENT_disableSENTPausePulse
sent.h	SENT_setCRCType
sent.h	SENT_setCRCWidth
sent.h	SENT_setDataNibble
sent.h	SENT_setTTClock
sent.h	SENT_setRxCRC
sent.h	SENT_enableDMAtrigger
sent.h	SENT_disableDMAtrigger
sent.h	SENT_configureFIFOMode
sent.h	SENT_setGFILTClockCycle
sent.h	SENT_enableTimeStamp
sent.h	SENT_disableTimeStamp
sent.h	SENT_enableERRDATAwrite
sent.h	SENT_disableERRDATAwrite
sent.h	SENT_enableMTPMode
sent.h	SENT_disableMTPMode
sent.h	SENT_setRFIFOTriggerLevel
RFDATA	
sent.h	SENT_getFIFO
RSDATA	
-	
RSTAT	
-	
RCFG2	
sent.h	SENT_enableDMAtrigger
sent.h	SENT_disableDMAtrigger
sent.h	SENT_configureFIFOMode
sent.h	SENT_setGFILTClockCycle
sent.h	SENT_enableTimeStamp
sent.h	SENT_disableTimeStamp
sent.h	SENT_enableERRDATAwrite
sent.h	SENT_disableERRDATAwrite
sent.h	SENT_enableMTPMode
sent.h	SENT_disableMTPMode
sent.h	SENT_setRFIFOTriggerLevel
RINTFLAG	

Table 42-8. SENT Registers to Driverlib Functions (continued)

File	Driverlib Function
sent.h	SENT_readInterruptStatus
REINT	
sent.h	SENT_enableInterrupt
sent.h	SENT_disableInterrupt
RSETINT	
sent.h	SENT_setSWInterrupt
RCLRINT	
sent.h	SENT_clearInterruptFlag
CSENT_SWR	
sent.h	SENT_enableSoftwareReset
sent.h	SENT_disableSoftwareReset
DATA0_MAP	
sent.h	SENT_setDataSortingFormat
DATA1_MAP	
-	
CSENT_TO	
sent.h	SENT_configureTimeout
CSENT_RXD	
sent.h	SENT_readSerialDataIn
RXVAL_CNT	
-	
RXDEDGE_CNT	
-	
SWR_RXVAL_CNT	
-	
SWR_RXDEDGE_CNT	
-	
CSENT_VERSION	
-	
MDATA(i)	
sent.h	SENT_getFrameData
BC_MTP_EN	
sent.h	SENT_enableMTPChannel
sent.h	SENT_disableMTPChannel
BC_MTP_CMP1	
sent.h	SENT_setBroadcastChMTPToggleTime
BC_MTP_CMP2	
-	
BC_MTP_CMP3	
-	
BC_MTP_CMP4	
-	
BC_MTP_CMP5	
-	
BC_MTP_CMP6	

Table 42-8. SENT Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
BC_MTP_CMP7	
-	
BC_MTP_CMP8	
-	
BC_MTP_CMP9	
-	
BC_MTP_PERIOD	
sent.h	SENT_setMTPPeriod
BC_TRIGSEL	
sent.h	SENT_setMTPTriggerSelect
BC_MTP_SWTR	
sent.h	SENT_enableMTPSoftwareTrigger
sent.h	SENT_disableMTPSoftwareTrigger
S1_MTP_EN	
-	
S1_MTP_CMP1	
sent.h	SENT_setSensor1MTPToggleTime
S1_MTP_CMP2	
-	
S1_MTP_CMP3	
-	
S1_MTP_CMP4	
-	
S1_MTP_CMP5	
-	
S1_MTP_CMP6	
-	
S1_MTP_CMP7	
-	
S1_MTP_CMP8	
-	
S1_MTP_CMP9	
-	
S1_MTP_CMP10RE	
-	
S1_MTP_PERIOD	
-	
S1_MTP_TO	
sent.h	SENT_configureMTPTimeout
S1_TRIGSEL	
-	
S1_MTP_SWTR	
-	
S2_MTP_EN	

Table 42-8. SENT Registers to Driverlib Functions (continued)

File	Driverlib Function
-	
S2_MTP_CMP1	
sent.h	SENT_setSensor2MTPToggleTime
S2_MTP_CMP2	
-	
S2_MTP_CMP3	
-	
S2_MTP_CMP4	
-	
S2_MTP_CMP5	
-	
S2_MTP_CMP6	
-	
S2_MTP_CMP7	
-	
S2_MTP_CMP8	
-	
S2_MTP_CMP9	
-	
S2_MTP_CMP10RE	
-	
S2_MTP_PERIOD	
-	
S2_MTP_TO	
-	
S2_TRIGSEL	
-	
S2_MTP_SWTR	
-	
S3_MTP_EN	
-	
S3_MTP_CMP1	
sent.h	SENT_setSensor3MTPToggleTime
S3_MTP_CMP2	
-	
S3_MTP_CMP3	
-	
S3_MTP_CMP4	
-	
S3_MTP_CMP5	
-	
S3_MTP_CMP6	
-	
S3_MTP_CMP7	
-	

Table 42-8. SENT Registers to Driverlib Functions (continued)

File	Driverlib Function
S3_MTP_CMP8	
-	
S3_MTP_CMP9	
-	
S3_MTP_CMP10RE	
-	
S3_MTP_PERIOD	
-	
S3_MTP_TO	
-	
S3_TRIGSEL	
-	
S3_MTP_SWTR	
-	
S4_MTP_EN	
-	
S4_MTP_CMP1	
sent.h	SENT_setSensor4MTPToggleTime
S4_MTP_CMP2	
-	
S4_MTP_CMP3	
-	
S4_MTP_CMP4	
-	
S4_MTP_CMP5	
-	
S4_MTP_CMP6	
-	
S4_MTP_CMP7	
-	
S4_MTP_CMP8	
-	
S4_MTP_CMP9	
-	
S4_MTP_CMP10RE	
-	
S4_MTP_PERIOD	
-	
S4_MTP_TO	
-	
S4_TRIGSEL	
-	
S4_MTP_SWTR	
-	
WAITTIME	

Table 42-8. SENT Registers to Driverlib Functions (continued)

File	Driverlib Function
sent.h	SENT_setMTPWaitTime
TPGENSTAT	
-	
MTP_VERSION	
-	
MTP_SWR	
sent.h	SENT_setMTPSoftwareReset

42.7.2 SENT Examples

NOTE: These examples are located in the C29SDK installation from ti.com at the following location: mcu_sdk_DEVICE_GPN/examples/driverlib/CORE_IF_MULTICORE/sent

Cloud access to these examples is available at the following link: dev.ti.com C29SDK Examples.

42.7.2.1 SENT Single Sensor - SINGLE_CORE

FILE: sent_ex1_single_sensor.c

This program will receive sensor data from a magnetic angle sensor using the SENT communication protocol. This data can be further utilized to convert and get the sensed information.

The sensor sends out a signal that is made up of a string of pulses with data encoded as falling to falling edge periods. It happens independently of any receiver module activity, which occurs without the receiver module sending a sync signal. The modulated signal with a constant amplitude voltage and an evaluation of the time interval between two falling edges (a single edge) is delivered in units of 4 bits (1 nibble), which can represent values ranging from 0 to 15.

This example configures SENT module to receive 6 Data-nibble per frame for fast channel.

External Connections

- Connect GPIO58 to sensor's SENT channel 1 pin

Watch Variables

- *SENT_Frame*[] - Frame received from the Sensor
- *timestamp*[] - Time stamp of the data nibble received

42.8 SENT Registers

This section describes the SENT Registers.

42.8.1 SENT Base Address Table

Table 42-9. SENT Base Address Table

Structure	DriverLib Name	Base Address	Frame Applicable	CPU1	CPU2	CPU3	RTDMA1	RTDMA2	HSM	Pipeline Protected
SENT_CFG	SENT1CSENT_BASE	0x6006_8000	YES	YES	YES	YES	YES	YES	-	YES
SENT_MEM	SENT1MEM_BASE	0x6006_8400	YES	YES	YES	YES	YES	YES	-	YES
SENT_MTPG	SENT1MTPG_BASE	0x6006_8800	YES	YES	YES	YES	YES	YES	-	YES
SENT_CFG	SENT2CSENT_BASE	0x6006_9000	YES	YES	YES	YES	YES	YES	-	YES
SENT_MEM	SENT2MEM_BASE	0x6006_9400	YES	YES	YES	YES	YES	YES	-	YES
SENT_MTPG	SENT2MTPG_BASE	0x6006_9800	YES	YES	YES	YES	YES	YES	-	YES
SENT_CFG	SENT3CSENT_BASE	0x6006_A000	YES	YES	YES	YES	YES	YES	-	YES
SENT_MEM	SENT3MEM_BASE	0x6006_A400	YES	YES	YES	YES	YES	YES	-	YES
SENT_MTPG	SENT3MTPG_BASE	0x6006_A800	YES	YES	YES	YES	YES	YES	-	YES
SENT_CFG	SENT4CSENT_BASE	0x6006_B000	YES	YES	YES	YES	YES	YES	-	YES
SENT_MEM	SENT4MEM_BASE	0x6006_B400	YES	YES	YES	YES	YES	YES	-	YES
SENT_MTPG	SENT4MTPG_BASE	0x6006_B800	YES	YES	YES	YES	YES	YES	-	YES
SENT_CFG	SENT5CSENT_BASE	0x6006_C000	YES	YES	YES	YES	YES	YES	-	YES
SENT_MEM	SENT5MEM_BASE	0x6006_C400	YES	YES	YES	YES	YES	YES	-	YES
SENT_MTPG	SENT5MTPG_BASE	0x6006_C800	YES	YES	YES	YES	YES	YES	-	YES
SENT_CFG	SENT6CSENT_BASE	0x6006_D000	YES	YES	YES	YES	YES	YES	-	YES
SENT_MEM	SENT6MEM_BASE	0x6006_D400	YES	YES	YES	YES	YES	YES	-	YES
SENT_MTPG	SENT6MTPG_BASE	0x6006_D800	YES	YES	YES	YES	YES	YES	-	YES

42.8.2 SENT_CFG Registers

Table 42-10 lists the memory-mapped registers for the SENT_CFG registers. All register offset addresses not listed in Table 42-10 should be considered as reserved locations and the register contents should not be modified.

Table 42-10. SENT_CFG Registers

Offset	Acronym	Register Name
20h	RCFG	Receiver Configuration Register
24h	RFDATA	FIFO's read data
28h	RSDATA	Receiver slow channel Data Register
2Ch	RSTAT	Receiver Status Register
34h	RCFG2	Receiver Configuration 2 Register
40h	RINTFLAG	Receiver Interrupt Status Register
44h	REINT	Receiver Interrupt Enable Register
48h	RSETINT	Receiver Set Interrupt Register
4Ch	RCLRINT	Receiver Clear Interrupt Register
50h	CSENT_SWR	CSENT Software Reset register
54h	DATA0_MAP	Receiver Data0 Sorting format
58h	DATA1_MAP	Receiver Data1 Sorting format register
5Ch	CSENT_TO	CSENT's Timeout register
60h	CSENT_RXD	CSENT Receiver Serial Data input register
64h	RXVAL_CNT	CSENT Rx Valid Frames Counter
68h	RXDEDGE_CNT	CSENT Receiver Serial Data Input Falling-edge Counter
6Ch	SWR_RXVAL_CNT	Valid CSENT frames counter Sofware Reset
70h	SWR_RXDEDGE_CNT	CSENT Receiver Serial Data input Falling-edge Counter Software Reset
7Ch	CSENT_VERSION	CSENT_XRF Version Number register

Complex bit access types are encoded to fit into small table cells. Table 42-11 shows the codes that are used for access types in this section.

Table 42-11. SENT_CFG Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

42.8.2.1 RCFG Register (Offset = 20h) [Reset = 012C0000h]

RCFG is shown in Figure 42-13 and described in Table 42-12.

Return to the Summary Table.

Receiver Configuration Register

Figure 42-13. RCFG Register

31	30	29	28	27	26	25	24
TTCLK							
R/W-12Ch							
23	22	21	20	19	18	17	16
TTCLK							
R/W-12Ch							
15	14	13	12	11	10	9	8
RSVD				RX_CRCTYPE	RX_CRC_WITH_STATUS	CRC_WIDTH	
R-0h				R/W-0h	R/W-0h	R/W-0h	
7	6	5	4	3	2	1	0
RX_DNIB				RX_PPENB	RSVD_0	RX_CRCENB	RX_ENB
R/W-0h				R/W-0h	R-0h	R/W-0h	R/W-0h

Table 42-12. RCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TTCLK	R/W	12Ch	The TTCLK determines the number of clock cycles per Tick when generating the internal tick clock. This tick clock is used for the initial calculation of calibration Sync errors (25% differential). The default is 300 (12Ch), which is typically 3 us ticks for 100 Mhz clock.
15-12	RSVD	R	0h	Reserved bits
11	RX_CRCTYPE	R/W	0h	0h: The recommended CRC (J2716 2010 or newer spec) checksum calculation is used. 1h: The original Legacy CRC (J2716 2007 spec) checksum calculation is used.
10	RX_CRC_WITH_STATUS	R/W	0h	0h: The CRC checksum calculation for the data nibbles only. 1h: The CRC checksum calculation include the status nibble and data nibble.
9-8	CRC_WIDTH	R/W	0h	CRC_WIDTH indicates the expected number of CRC bits for fast channel. The default is 0h, which is a 4-bit CRC. 0h: 4-bit CRC 1h: 6-bit CRC 2h: 8-bit CRC 3h: Reserved
7-4	RX_DNIB	R/W	0h	RX_DNIB indicates the number of data nibbles should be expected (1 to 6) for standard SENT. Some of the enhanced sensors support 7 and 8 nibbles. RX_DNIB[3:0] 0h: Reserved 1h: 1 data nibble per-frame 2h: 2 data nibbles per-frame 3h: 3 data nibbles per-frame 4h: 4 data nibbles per-frame 5h: 5 data nibbles per-frame 6h: 6 data nibbles per-frame 7h: 7 data nibbles per-frame 8h: 8 data nibbles per-frame 9h-Fh: Reserved

Table 42-12. RCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	RX_PPENB	R/W	0h	0h: The sensor is continuously supplying SENT data that the calibration pulse follows immediately after the CRC. 1h: Indicates that Pause Pulse will be expected after the CRC if this is enabled. The RX_PPENB must be enabled for frequency drift's detection on the fixed length frames.
2	RSVD_0	R	0h	Reserve bit
1	RX_CRCENB	R/W	0h	0h: The CRC calculation and check is disabled 1h: The CRC for fast channel and slow channel is enabled to calculate and check.
0	RX_ENB	R/W	0h	0h: CSENT Receiver is disabled 1h: The receiver is enabled to monitor the SENT bus. When using a master trigger pulse to trigger a sensor, the RX_ENB must be set to 0 and the CSENT receiver is controlled by MTPG module.

42.8.2.2 RFDATA Register (Offset = 24h) [Reset = 00000000h]

RFDATA is shown in Figure 42-14 and described in Table 42-13.

Return to the Summary Table.

FIFO's read data

Figure 42-14. RFDATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFO_RDATA																															
R-0h																															

Table 42-13. RFDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FIFO_RDATA	R	0h	FIFO_RDATA contains the received CSENT fast channel data located at the top of the receive FIFO. The read data can be timestamp or data0 or data1.

42.8.2.3 RSDATA Register (Offset = 28h) [Reset = 00000000h]

RSDATA is shown in Figure 42-15 and described in Table 42-14.

Return to the Summary Table.

Receiver slow channel Data Register

Figure 42-15. RSDATA Register

31	30	29	28	27	26	25	24
MESSAGEID							
R-0h							
23	22	21	20	19	18	17	16
DATA							
R-0h							
15	14	13	12	11	10	9	8
DATA							
R-0h							
7	6	5	4	3	2	1	0
C	RSVD	CRC					
R-0h	R-0h	R-0h					

Table 42-14. RSDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MESSAGEID	R	0h	Received slow channel Message ID. 4-bit Received Message ID for Short Serial Message or Enhanced 16-bit Serial Message. 8-bit Received Message ID for Enhanced 12-bit Serial Message.
23-8	DATA	R	0h	Received slow channel Data. 8-bit data Received Short Serial Message, or 12-bit/16-bit data Received Enhanced Serial Message.
7	C	R	0h	Received slow channel Select type data 0h: Enhanced 12-bit Serial Message Data 1h: Enhanced 16-bit Serial Message Data
6	RSVD	R	0h	Reserved bit
5-0	CRC	R	0h	Received slow channel CRC data. 4-bit CRC for Received Short Serial Message or 6-bit CRC for Reveived Enhanced Serial Message.

42.8.2.4 RSTAT Register (Offset = 2Ch) [Reset = 00000000h]

RSTAT is shown in Figure 42-16 and described in Table 42-15.

Return to the Summary Table.

Receiver Status Register

Figure 42-16. RSTAT Register

31	30	29	28	27	26	25	24
RX_TTCLK							
R-0h							
23	22	21	20	19	18	17	16
RX_TTCLK							
R-0h							
15	14	13	12	11	10	9	8
RX_SYNC	RXSLOW_DN	RXSLOW_ACT	RXSLOW_ST				
R-0h	R-0h	R-0h	R-0h				
7	6	5	4	3	2	1	0
RSVD			PP	RXNIB			
R-0h			R-0h	R-0h			

Table 42-15. RSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RX_TTCLK	R	0h	RX_TTCLK contains the most recent synchronization tick time calculation in clocks. The user can read this value and compare it to the previous frame to determine if the tick clock time has changed significantly.
15	RX_SYNC	R	0h	0h: Indicates that the Receiver is receiving data 1h: Indicates that the Receiver is waiting for a synchronization period
14	RXSLOW_DN	R	0h	0h: No slow channel serial message has been received 1h: Indicates that the receiving of the slow channel serial message is done. This bit will be cleared after the received RSDATA is read
13	RXSLOW_ACT	R	0h	0h: Indicates that the slow channel is not active 1h: Indicates that the receiving of the slow channel serial message is active
12-8	RXSLOW_ST	R	0h	RXSLOW_ST contains the receiving status of the slow channel data (Message ID or Data or CRC) during receiving Status and Communication nibble.
7-5	RSVD	R	0h	Reserved bits
4	PP	R	0h	0h: No pause pulse 1h: Indicates that it is currently receiving the pause pulse (only if the RX_PPENB in RCFG register is set to 1)
3-0	RXNIB	R	0h	RXNIB indicates Received data nibble's status. RXNIB[3:0] 0h: Invalid 1h: Receiving Data Nibble 1 2h: Receiving Data Nibble 2 3h: Receiving Data Nibble 3 4h: Receiving Data Nibble 4 5h: Receiving Data Nibble 5 6h: Receiving Data Nibble 6 7h: Receiving Data Nibble 7 8h: Receiving Data Nibble 8 9h-Fh: Invalid

42.8.2.5 RCFG2 Register (Offset = 34h) [Reset = 00000050h]

RCFG2 is shown in Figure 42-17 and described in Table 42-16.

Return to the Summary Table.

Receiver Configuration 2 Register

Figure 42-17. RCFG2 Register

31	30	29	28	27	26	25	24
RSVD_0							
R-0h							
23	22	21	20	19	18	17	16
RSVD_0				RFIFO_TRIGLEV			
R-0h				R/W-0h			
15	14	13	12	11	10	9	8
RSVD			MTP_MODE	GFILT			
R-0h			R/W-0h	R/W-5h			
7	6	5	4	3	2	1	0
GFILT				NOWR_ERRDATA	TSTAMP_DIS	DMA_ENB	FIFO_MODE
R/W-5h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 42-16. RCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RSVD_0	R	0h	Reserved bits
19-16	RFIFO_TRIGLEV	R/W	0h	This field indicates the trigger level for the receiver FIFO trigger interrupt and the DMA trigger. The level can be from 1 up to the maximum level of the FIFOs depth (16). This field is used only when the FIFO_MODE is 1. Receiver FIFO trigger interrupt will be disabled if RFIFO_TRIGLEV is set to 0 (default value).
15-13	RSVD	R	0h	Reserved bits
12	MTP_MODE	R/W	0h	0h: MTP_MODE is disabled 1h: This mode indicates that the Receiver will send the master trigger pulse(s) and then expects Sensor's data response
11-4	GFILT	R/W	5h	GFILT indicates the number of clock cycle pulse on the Receiver Serial Data input that will be filtered out. A range from 0 (bypass) to 255 can be specified. The default is 5.
3	NOWR_ERRDATA	R/W	0h	0h: Default value, indicates always write to the memory although the data contains errors 1h: The received message with errors will not be written to the FIFO or memory
2	TSTAMP_DIS	R/W	0h	0h: Default value that indicates the timestamp will be written to the memory. The Timestamp will be recorded at the start of the incoming frame 1h: Disable the timestamp. The timestamp of the frame received will not be written into the memory
1	DMA_ENB	R/W	0h	0h: Disable DMA trigger 1h: Enable DMA trigger. The receiver FIFO DMA trigger is generated when the FIFO level is >= trigger level (RFIFO_TRIGLEV)

Table 42-16. RCFG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	FIFO_MODE	R/W	0h	0h: The Timestamp, data0 and data1 data will be mapped directly to a memory. For Multiple Sensor mode, the timestamp and data for each sensor will be written into the same FIFO/memory. The default is 0 for direct mapping 1h: The Receivers memory is in FIFO mode. It contains Timestamp, data0, and data1. The FIFO width is 32-bit and its depth is configurable

42.8.2.6 RINTFLAG Register (Offset = 40h) [Reset = 00000000h]

RINTFLAG is shown in Figure 42-18 and described in Table 42-17.

Return to the Summary Table.

Receiver Interrupt Status Register

Figure 42-18. RINTFLAG Register

31	30	29	28	27	26	25	24
RSVD	OVFTRIG_ERR					RFIFO_UNDFERR	RFIFO_OVFERR
R-0h	R-0h					R-0h	R-0h
23	22	21	20	19	18	17	16
RFIFO_TRIGGER	FREQDRIFT_ERR	RFAST_SYNCERR	RFAST_SYNCERR25	RTIMEOUT_ERR			
R-0h	R-0h	R-0h	R-0h	R-0h			
15	14	13	12	11	10	9	8
RFAST_S4FRME	RFAST_S3FRME	RFAST_S2FRME	RFAST_S1FRME	RFAST_S4CRCE	RFAST_S3CRCE	RFAST_S2CRCE	RFAST_S1CRCE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RSLOW_FORMATERR	RSLOW_CRCERR	RSLOW_DV	RFAST_S4DV	RFAST_S3DV	RFAST_S2DV	RFAST_S1DV	GLBL
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 42-17. RINTFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RSVD	R	0h	Reserved bit
30-26	OVFTRIG_ERR	R	0h	OVFTRIG_ERR: when one of the bit is set, it indicates that there is an overflow trigger on that channel. This error appears only when the MTP_MODE is enabled. It happens when the same channel trigger pulse generator receives trigger request while it is busy. OVFTRIG_ERR[4:0] 00h: No Overflow Trigger 01h: Overflow Trigger on the Broadcast channel 02h: Overflow Trigger on the Sensor 1 channel 04h: Overflow Trigger on the Sensor 2 channel 08h: Overflow Trigger on the Sensor 3 channel 10h: Overflow Trigger on the Sensor 4 channel
25	RFIFO_UNDFERR	R	0h	0h: No underflow error on the receiver FIFO 1h: The receiver FIFO is underflow. The CPU reads the receiver FIFO when it is empty
24	RFIFO_OVFERR	R	0h	0h: No overflow error on the receiver FIFO 1h: The Receiver FIFO is overflow, a write to the receiver FIFO while it is full
23	RFIFO_TRIGGER	R	0h	0h: No FIFO Trigger 1h: The number of data in the Receivers FIFO is >= the programmed trigger
22	FREQDRIFT_ERR	R	0h	0h: No frequency drift error 1h: There is a frequency drifts error on the fixed length frames. The ratio of the calibration pulse to the message length varies by > 1/64 or < -1/64 from one message to another . The RX_PPENB in RCFG register is required to be enabled for the fixed frames length
21	RFAST_SYNCERR	R	0h	0h: No synchronization error 1h: Synchronization error, the successive calibration pulses differ by >+1.5625% or < -1.5625%

Table 42-17. RINTFLAG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	RFAST_SYNCERR25	R	0h	0h: No Calibration/Synchonization error 1h: Indicates that the Calibration/Synchronization pulse varies by more than 25% from the expected 56 clock ticks
19-16	RTIMEOUT_ERR	R	0h	RTIMEOUT_ERR is used for the Synchronous mode Sensor, which needs master trigger pulse (MTP_MODE is enabled) or standard CSENT (MTP_MODE is disabled). For the MTP_MODE, this flag is set when the receiver does receive the low Synchronous pulse in reponse to a trigger pulse before the timeout specified. RTIMEOUT_ERR[3:0] 0h: No Timeout 1h: No response from Sensor 1 in timeout specified 2h: No response from Sensor 2 in timeout specified 4h: No response from Sensor 3 in timeout specified 8h: No response from Sensor 4 in timeout specified For standard CSENT (MTP_MODE is disabled) timeout, RTIMEOUT_ERR[0] is set when the receiver does not receive the next frame's low synchronous pulse after a frame is received (long idle time) before the timeout specified.
15	RFAST_S4FRME	R	0h	0h: No Frame error in the Sensor 4 fast channel data 1h: For multi-sensor mode, it indicates that there is a Frame Error (invalid nibble) in the Sensor 4 fast channel data
14	RFAST_S3FRME	R	0h	0h: No Frame error in the Sensor 3 fast channel data 1h: For multi-sensor mode, it indicates that there is a Frame Error (invalid nibble) in the Sensor 3 fast channel data
13	RFAST_S2FRME	R	0h	0h: No Frame error in the Sensor 2 fast channel data 1h: For multi-sensor mode, it indicates that there is a Frame Error (invalid nibble) in the Sensor 2 fast channel data
12	RFAST_S1FRME	R	0h	0h: No Frame error in the Sensor 1 fast channel data 1h: For non MTP_MODE sensor (standard CSENT), it indicates that there is a Frame Error (invalid nibble) in the fast channel data. In the multi-sensor mode, it indicates that there is a Frame error in the Sensor 1 fast channel data
11	RFAST_S4CRCE	R	0h	0h: No CRC Error in the Sensor 4 fast channel data 1h: For multi-sensor mode, it indicates that there is a CRC Error in the Sensor 4 fast channel data.
10	RFAST_S3CRCE	R	0h	0h: No CRC Error in the Sensor 3 fast channel data 1h: For multi-sensor mode, it indicates that there is a CRC Error in the Sensor 3 fast channel data
9	RFAST_S2CRCE	R	0h	0h: No CRC Error in the Sensor 2 fast channel data 1h: For multi-sensor mode, it indicates that there is a CRC Error in the Sensor 2 fast channel data
8	RFAST_S1CRCE	R	0h	0h: No Sensor 1 fast channel CRC error 1h: For non MTP_MODE sensor (standard CSENT), this flag will be set when there is a CRC error (mismatched CRC calculation result with received CRC data) in the received fast channel data. In multi-sensor mode, this will only be set when Sensor 1 fast channel data has CRC error
7	RSLOW_FORMATERR	R	0h	0h: No slow channel Format Error 1h: There is Format Error in the slow channel data
6	RSLOW_CRCERR	R	0h	0h: No slow channel CRC Error 1h: There is CRC Error in the slow channel data
5	RSLOW_DV	R	0h	0h: No valid slow channel data 1h: A valid slow channel data is received from single sensor
4	RFAST_S4DV	R	0h	0h: No valid Sensor 4 data is received 1h: In multi-sensor mode, it indicates Sensor 4 fast channel data is received
3	RFAST_S3DV	R	0h	0h: No valid Sensor 3 data is received 1h: In multi-sensor mode, it indicates Sensor 3 fast channel data is received

Table 42-17. RINTFLAG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RFAST_S2DV	R	0h	0h: No valid Sensor 2 data is received 1h: In multi-sensor mode, it indicates Sensor 2 fast channel data is received
1	RFAST_S1DV	R	0h	0h: No valid Sensor 1 data is received 1h: For non MTP_MODE sensor (standard CSENT) mode, it indicates a valid fast frame is received. In multi-sensor mode, it indicates Sensor 1 fast channel data is received
0	GLBL	R	0h	This GLBL flag will be set whenever an interrupt occurs (that is not masked)

42.8.2.7 REINT Register (Offset = 44h) [Reset = 00000000h]

REINT is shown in Figure 42-19 and described in Table 42-18.

Return to the Summary Table.

Receiver Interrupt Enable Register

Figure 42-19. REINT Register

31	30	29	28	27	26	25	24
RSVD_1	OVFTRIG_ERR_E					RFIFO_UNDFERR_E	RFIFO_OVFERR_E
R-0h	R/W-0h					R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RFIFO_TRIGGER_E	FREQDRIFT_ERR_E	RFAST_SYNCERR_E	RFAST_SYNCERR25_E	RTIMEOUT_ERR_E			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h			
15	14	13	12	11	10	9	8
RFAST_S4FRME_E	RFAST_S3FRME_E	RFAST_S2FRME_E	RFAST_S1FRME_E	RFAST_S4CRCE_E	RFAST_S3CRCE_E	RFAST_S2CRCE_E	RFAST_S1CRCE_E
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RSLOW_FORMATERR_E	RSLOW_CRCERR_E	RSLOW_DV_E	RFAST_S4DV_E	RFAST_S3DV_E	RFAST_S2DV_E	RFAST_S1DV_E	RSVD
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 42-18. REINT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RSVD_1	R	0h	Reserved bit
30-26	OVFTRIG_ERR_E	R/W	0h	00h: Interrupt disabled for overflow trigger error events 01h: Interrupt Enable for Broadcast Channel overflow trigger error event 02h: Interrupt Enable for Sensor 1 Channel overflow trigger error event 04h: Interrupt Enable for Sensor 2 Channel overflow trigger error event 08h: Interrupt Enable for Sensor 3 Channel overflow trigger error event 10h: Interrupt Enable for Sensor 4 Channel overflow trigger error event
25	RFIFO_UNDFERR_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for FIFO underflow error event
24	RFIFO_OVFERR_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for FIFO Overflow Error event
23	RFIFO_TRIGGER_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for received FIFO trigger event
22	FREQDRIFT_ERR_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for received frequency drift error event
21	RFAST_SYNCERR_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for received the successive calibration pulses error (differ by >+1.5625% or < -1.5625%) event
20	RFAST_SYNCERR25_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for received fast channel Synchronization Pulse indicating that it varied more than 25% from the expected 56 clock ticks

Table 42-18. REINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-16	RTIMEOUT_ERR_E	R/W	0h	0h: Disable Timeout Error Events 1h: Interrupt Enable for Sensor 1 Channel timeout error event 2h: Interrupt Enable for Sensor 2 Channel timeout error event 4h: Interrupt Enable for Sensor 3 Channel timeout error event 8h: Interrupt Enable for Sensor 4 Channel timeout error event
15	RFAST_S4FRME_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for received Sensor 4 fast channel Data with Frame Error event
14	RFAST_S3FRME_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for received Sensor 3 fast channel Data with Frame Error event
13	RFAST_S2FRME_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for received Sensor 2 fast channel Data with Frame Error event
12	RFAST_S1FRME_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for received Sensor 1 fast channel Data with Frame Error event
11	RFAST_S4CRCE_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for received Sensor 4 fast channel Data with CRC Error event
10	RFAST_S3CRCE_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for received Sensor 3 fast channel Data with CRC Error event
9	RFAST_S2CRCE_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for received Sensor 2 fast channel Data with CRC Error event
8	RFAST_S1CRCE_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for received Sensor 1 fast channel Data with CRC Error event
7	RSLOW_FORMATERR_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for Received slow channel Data with Format Error event
6	RSLOW_CRCERR_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for Received slow channel Data with CRC Error event
5	RSLOW_DV_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for Received slow channel Valid Data event
4	RFAST_S4DV_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for Received Sensor 4's fast channel Valid Data event
3	RFAST_S3DV_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for Received Sensor 3's fast channel Valid Data event
2	RFAST_S2DV_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for Received Sensor 2's fast channel Valid Data event
1	RFAST_S1DV_E	R/W	0h	0h: Interrupt disable 1h: Interrupt Enable for Received Sensor 1's fast channel Valid Data event
0	RSVD	R	0h	Reserved bit

42.8.2.8 RSETINT Register (Offset = 48h) [Reset = 00000000h]

RSETINT is shown in Figure 42-20 and described in Table 42-19.

Return to the Summary Table.

Receiver Set Interrupt Register

Figure 42-20. RSETINT Register

31	30	29	28	27	26	25	24
RSVD_0	OVFTRIG_ERR_S					RFIFO_UNDFERR_S	RFIFO_OVFERR_S
W-0h	W-0h					W-0h	W-0h
23	22	21	20	19	18	17	16
RFIFO_TRIGGER_S	FREQDRIFT_ERR_S	RFAST_SYNCERR_S	RFAST_SYNCERR25_S	RTIMEOUT_ERR_S			
W-0h	W-0h	W-0h	W-0h	W-0h			
15	14	13	12	11	10	9	8
RFAST_S4FRME_S	RFAST_S3FRME_S	RFAST_S2FRME_S	RFAST_S1FRME_S	RFAST_S4CRCE_S	RFAST_S3CRCE_S	RFAST_S2CRCE_S	RFAST_S1CRCE_S
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RSLOW_FORMATERR_S	RSLOW_CRCERR_S	RSLOW_DV_S	RFAST_S4DV_S	RFAST_S3DV_S	RFAST_S2DV_S	RFAST_S1DV_S	RSVD
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 42-19. RSETINT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RSVD_0	W	0h	Reserved bit
30-26	OVFTRIG_ERR_S	W	0h	00h: No interrupt set 01h: Software interrupt set Broadcast Overflow Trigger flag 02h: Software interrupt set Sensor 1 Overflow Trigger flag 04h: Software interrupt set Sensor 2 Overflow Trigger flag 08h: Software interrupt set Sensor 3 Overflow Trigger flag 10h: Software interrupt set Sensor 4 Overflow Trigger flag
25	RFIFO_UNDFERR_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFIFO_UNDFERR] flag
24	RFIFO_OVFERR_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFIFO_OVFERR] flag
23	RFIFO_TRIGGER_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFIFO_TRIGGER] flag
22	FREQDRIFT_ERR_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[FREQDRIFT_ERR] flag
21	RFAST_SYNCERR_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFAST_SYNCERR] flag
20	RFAST_SYNCERR25_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFAST_SYNCERR25] flag
19-16	RTIMEOUT_ERR_S	W	0h	0h: No interrupt set 1h: Software interrupt set Sensor 1 Timeout Error flag 2h: Software interrupt set Sensor 2 Timeout Error flag 4h: Software interrupt set Sensor 3 Timeout Error flag 8h: Software interrupt set Sensor 4 Timeout Error flag
15	RFAST_S4FRME_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFAST_S4FRME] flag
14	RFAST_S3FRME_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFAST_S3FRME] flag

Table 42-19. RSETINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	RFAST_S2FRME_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFAST_S2FRME] flag
12	RFAST_S1FRME_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFAST_S1FRME] flag
11	RFAST_S4CRCE_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFAST_S4CRCE] flag
10	RFAST_S3CRCE_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFAST_S3CRCE] flag
9	RFAST_S2CRCE_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFAST_S2CRCE] flag
8	RFAST_S1CRCE_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFAST_S1CRCE] flag
7	RSLOW_FORMATERR_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RSLOW_FORMATERR] flag
6	RSLOW_CRCERR_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RSLOW_CRCERR] flag
5	RSLOW_DV_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RSLOW_DV] flag
4	RFAST_S4DV_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFAST_S4DV] flag
3	RFAST_S3DV_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFAST_S3DV] flag
2	RFAST_S2DV_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFAST_S2DV] flag
1	RFAST_S1DV_S	W	0h	0h: No interrupt set 1h: Software interrupt set RINTFLAG[RFAST_S1DV] flag
0	RSVD	W	0h	Reserved bit

42.8.2.9 RCLRINT Register (Offset = 4Ch) [Reset = 00000000h]

RCLRINT is shown in Figure 42-21 and described in Table 42-20.

Return to the Summary Table.

Receiver Clear Interrupt Register

Figure 42-21. RCLRINT Register

31	30	29	28	27	26	25	24
RSVD	OVFTRIG_ERR_CLR					RFIFO_UNDFERR_CLR	RFIFO_OVFERR_CLR
W-0h	W-0h					W-0h	W-0h
23	22	21	20	19	18	17	16
RFIFO_TRIGGER_CLR	FREQDRIFT_ERR_CLR	RFAST_SYNCERR_CLR	RFAST_SYNCERR25_CLR	RTIMEOUT_ERR_CLR			
W-0h	W-0h	W-0h	W-0h	W-0h			
15	14	13	12	11	10	9	8
RFAST_S4FRME_CLR	RFAST_S3FRME_CLR	RFAST_S2FRME_CLR	RFAST_S1FRME_CLR	RFAST_S4CRCE_CLR	RFAST_S3CRCE_CLR	RFAST_S2CRCE_CLR	RFAST_S1CRCE_CLR
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RSLOW_FORMATERR_CLR	RSLOW_CRCERR_CLR	RSLOW_DV_CLR	RFAST_S4DV_CLR	RFAST_S3DV_CLR	RFAST_S2DV_CLR	RFAST_S1DV_CLR	GLBL_CLR
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 42-20. RCLRINT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RSVD	W	0h	Reserved bit
30-26	OVFTRIG_ERR_CLR	W	0h	00h: No Clear on Overflow Trigger Error interrupt flags 01h: Clear Broadcast's Overflow Trigger Error interrupt flag 02h: Clear Sensor 1 Overflow Trigger Error interrupt flag 04h: Clear Sensor 2 Overflow Trigger Error interrupt flag 08h: Clear Sensor 3 Overflow Trigger Error interrupt flag 10h: Clear Sensor 4 Overflow Trigger Error interrupt flag
25	RFIFO_UNDFERR_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFIFO_UNDFERR] interrupt flag
24	RFIFO_OVFERR_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFIFO_OVFERR] interrupt flag
23	RFIFO_TRIGGER_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFIFO_TRIGGER] interrupt flag
22	FREQDRIFT_ERR_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[FREQDRIFT_ERR] interrupt flag
21	RFAST_SYNCERR_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFAST_SYNCERR] interrupt flag
20	RFAST_SYNCERR25_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFAST_SYNCERR25] interrupt flag
19-16	RTIMEOUT_ERR_CLR	W	0h	0h: No clear on timeout interrupt flags 1h: Clear Sensor 1 Timeout Error flag 2h: Clear Sensor 2 Timeout Error flag 4h: Clear Sensor 3 Timeout Error flag 8h: Clear Sensor 4 Timeout Error flag
15	RFAST_S4FRME_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFAST_S4FRME] interrupt flag
14	RFAST_S3FRME_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFAST_S3FRME] interrupt flag

Table 42-20. RCLRINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	RFAST_S2FRME_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFAST_S2FRME] interrupt flag
12	RFAST_S1FRME_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFAST_S1FRME] interrupt flag
11	RFAST_S4CRCE_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFAST_S4CRCE] interrupt flag
10	RFAST_S3CRCE_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFAST_S3CRCE] interrupt flag
9	RFAST_S2CRCE_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFAST_S2CRCE] interrupt flag
8	RFAST_S1CRCE_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFAST_S1CRCE] interrupt flag
7	RSLOW_FORMATERR_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RSLOW_FORMATERR] interrupt flag
6	RSLOW_CRCERR_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RSLOW_CRCERR] interrupt flag
5	RSLOW_DV_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RSLOW_DV] interrupt flag
4	RFAST_S4DV_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFAST_S4DV] interrupt flag
3	RFAST_S3DV_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFAST_S3DV] interrupt flag
2	RFAST_S2DV_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFAST_S2DV] interrupt flag
1	RFAST_S1DV_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[RFAST_S1DV] interrupt flag
0	GLBL_CLR	W	0h	0h: No clear 1h: Clear RINTFLAG[GLBL] flag

42.8.2.10 CSENT_SWR Register (Offset = 50h) [Reset = 00000000h]

CSENT_SWR is shown in Figure 42-22 and described in Table 42-21.

Return to the Summary Table.

CSENT Software Reset register

Figure 42-22. CSENT_SWR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD															SWR
W-0h															W-0h

Table 42-21. CSENT_SWR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RSVD	W	0h	Reserved bits
0	SWR	W	0h	0h: No CSENT software reset 1h: It will cause CSENT Receiver software reset and all registers will have their default value.

42.8.2.11 DATA0_MAP Register (Offset = 54h) [Reset = 00000000h]

DATA0_MAP is shown in Figure 42-23 and described in Table 42-22.

Return to the Summary Table.

Receiver Data0 Sorting format

Figure 42-23. DATA0_MAP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
D0_NIB7				D0_NIB6				D0_NIB5				D0_NIB4			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D0_NIB3				D0_NIB2				D0_NIB1				D0_NIB0			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 42-22. DATA0_MAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	D0_NIB7	R/W	0h	Data0 nibble 7 sorting format register. This field configures where the received nibbles are stored in the data0 map for FIFO or direct map mode. D0_NIB7[3:0] 0h: None 1h: Data nibble 1 2h: Data nibble 2 3h: Data nibble 3 4h: Data nibble 4 5h: Data nibble 5 6h: Data nibble 6 7h: Data nibble 7 8h: Data nibble 8 9h: Reserved Ah: CRC nibble 1 Bh: CRC nibble 2 Ch-Eh: Reserved Fh: Status
27-24	D0_NIB6	R/W	0h	Data0 nibble 6 sorting format register. This field configures where the received nibbles are stored in the data0 map for FIFO or direct map mode. D0_NIB6[3:0] 0h: None 1h: Data nibble 1 2h: Data nibble 2 3h: Data nibble 3 4h: Data nibble 4 5h: Data nibble 5 6h: Data nibble 6 7h: Data nibble 7 8h: Data nibble 8 9h: Reserved Ah: CRC nibble 1 Bh: CRC nibble 2 Ch-Eh: Reserved Fh: Status

Table 42-22. DATA0_MAP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-20	D0_NIB5	R/W	0h	Data0 nibble 5 sorting format register. This field configures where the received nibbles are stored in the data0 map for FIFO or direct map mode. D0_NIB5[3:0] 0h: None 1h: Data nibble 1 2h: Data nibble 2 3h: Data nibble 3 4h: Data nibble 4 5h: Data nibble 5 6h: Data nibble 6 7h: Data nibble 7 8h: Data nibble 8 9h: Reserved Ah: CRC nibble 1 Bh: CRC nibble 2 Ch-Eh: Reserved Fh: Status
19-16	D0_NIB4	R/W	0h	Data0 nibble 4 sorting format register. This field configures where the received nibbles are stored in the data0 map for FIFO or direct map mode. D0_NIB4[3:0] 0h: None 1h: Data nibble 1 2h: Data nibble 2 3h: Data nibble 3 4h: Data nibble 4 5h: Data nibble 5 6h: Data nibble 6 7h: Data nibble 7 8h: Data nibble 8 9h: Reserved Ah: CRC nibble 1 Bh: CRC nibble 2 Ch-Eh: Reserved Fh: Status
15-12	D0_NIB3	R/W	0h	Data0 nibble 3 sorting format register. This field configures where the received nibbles are stored in the data0 map for FIFO or direct map mode. D0_NIB3[3:0] 0h: None 1h: Data nibble 1 2h: Data nibble 2 3h: Data nibble 3 4h: Data nibble 4 5h: Data nibble 5 6h: Data nibble 6 7h: Data nibble 7 8h: Data nibble 8 9h: Reserved Ah: CRC nibble 1 Bh: CRC nibble 2 Ch-Eh: Reserved Fh: Status

Table 42-22. DATA0_MAP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-8	D0_NIB2	R/W	0h	Data0 nibble 2 sorting format register. This field configures where the received nibbles are stored in the data0 map for FIFO or direct map mode. D0_NIB2[3:0] 0h: None 1h: Data nibble 1 2h: Data nibble 2 3h: Data nibble 3 4h: Data nibble 4 5h: Data nibble 5 6h: Data nibble 6 7h: Data nibble 7 8h: Data nibble 8 9h: Reserved Ah: CRC nibble 1 Bh: CRC nibble 2 Ch-Eh: Reserved Fh: Status
7-4	D0_NIB1	R/W	0h	Data0 nibble 1 sorting format register. This field configures where the received nibbles are stored in the data0 map for FIFO or direct map mode. D0_NIB1[3:0] 0h: None 1h: Data nibble 1 2h: Data nibble 2 3h: Data nibble 3 4h: Data nibble 4 5h: Data nibble 5 6h: Data nibble 6 7h: Data nibble 7 8h: Data nibble 8 9h: Reserved Ah: CRC nibble 1 Bh: CRC nibble 2 Ch-Eh: Reserved Fh: Status
3-0	D0_NIB0	R/W	0h	Data0 nibble 0 sorting format register. This field configures where the received nibbles are stored in the data0 map for FIFO or direct map mode. D0_NIB0[3:0] 0h: None 1h: Data nibble 1 2h: Data nibble 2 3h: Data nibble 3 4h: Data nibble 4 5h: Data nibble 5 6h: Data nibble 6 7h: Data nibble 7 8h: Data nibble 8 9h: Reserved Ah: CRC nibble 1 Bh: CRC nibble 2 Ch-Eh: Reserved Fh: Status

42.8.2.12 DATA1_MAP Register (Offset = 58h) [Reset = 00000000h]

DATA1_MAP is shown in Figure 42-24 and described in Table 42-23.

Return to the Summary Table.

Receiver Data1 Sorting format register

Figure 42-24. DATA1_MAP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
D1_NIB7				D1_NIB6				D1_NIB5				D1_NIB4			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D1_NIB3				D1_NIB2				D1_NIB1				D1_NIB0			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 42-23. DATA1_MAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	D1_NIB7	R/W	0h	Data1 nibble 7 sorting format register. This field configures where the received nibbles are stored in the data1 map for FIFO or direct map mode. D1_NIB7[3:0] 0h: None 1h: Data nibble 1 2h: Data nibble 2 3h: Data nibble 3 4h: Data nibble 4 5h: Data nibble 5 6h: Data nibble 6 7h: Data nibble 7 8h: Data nibble 8 9h: Reserved Ah: CRC nibble 1 Bh: CRC nibble 2 Ch-Eh: Reserved Fh: Status
27-24	D1_NIB6	R/W	0h	Data1 nibble 6 sorting format register. This field configures where the received nibbles are stored in the data1 map for FIFO or direct map mode. D1_NIB6[3:0] 0h: None 1h: Data nibble 1 2h: Data nibble 2 3h: Data nibble 3 4h: Data nibble 4 5h: Data nibble 5 6h: Data nibble 6 7h: Data nibble 7 8h: Data nibble 8 9h: Reserved Ah: CRC nibble 1 Bh: CRC nibble 2 Ch-Eh: Reserved Fh: Status

Table 42-23. DATA1_MAP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-20	D1_NIB5	R/W	0h	Data1 nibble 5 sorting format register. This field configures where the received nibbles are stored in the data1 map for FIFO or direct map mode. D1_NIB5[3:0] 0h: None 1h: Data nibble 1 2h: Data nibble 2 3h: Data nibble 3 4h: Data nibble 4 5h: Data nibble 5 6h: Data nibble 6 7h: Data nibble 7 8h: Data nibble 8 9h: Reserved Ah: CRC nibble 1 Bh: CRC nibble 2 Ch-Eh: Reserved Fh: Status
19-16	D1_NIB4	R/W	0h	Data1 nibble 4 sorting format register. This field configures where the received nibbles are stored in the data1 map for FIFO or direct map mode. D1_NIB4[3:0] 0h: None 1h: Data nibble 1 2h: Data nibble 2 3h: Data nibble 3 4h: Data nibble 4 5h: Data nibble 5 6h: Data nibble 6 7h: Data nibble 7 8h: Data nibble 8 9h: Reserved Ah: CRC nibble 1 Bh: CRC nibble 2 Ch-Eh: Reserved Fh: Status
15-12	D1_NIB3	R/W	0h	Data1 nibble 3 sorting format register. This field configures where the received nibbles are stored in the data1 map for FIFO or direct map mode. D1_NIB3[3:0] 0h: None 1h: Data nibble 1 2h: Data nibble 2 3h: Data nibble 3 4h: Data nibble 4 5h: Data nibble 5 6h: Data nibble 6 7h: Data nibble 7 8h: Data nibble 8 9h: Reserved Ah: CRC nibble 1 Bh: CRC nibble 2 Ch-Eh: Reserved Fh: Status

Table 42-23. DATA1_MAP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-8	D1_NIB2	R/W	0h	Data1 nibble 2 sorting format register. This field configures where the received nibbles are stored in the data1 map for FIFO or direct map mode. D1_NIB2[3:0] 0h: None 1h: Data nibble 1 2h: Data nibble 2 3h: Data nibble 3 4h: Data nibble 4 5h: Data nibble 5 6h: Data nibble 6 7h: Data nibble 7 8h: Data nibble 8 9h: Reserved Ah: CRC nibble 1 Bh: CRC nibble 2 Ch-Eh: Reserved Fh: Status
7-4	D1_NIB1	R/W	0h	Data1 nibble 1 sorting format register. This field configures where the received nibbles are stored in the data1 map for FIFO or direct map mode. D1_NIB1[3:0] 0h: None 1h: Data nibble 1 2h: Data nibble 2 3h: Data nibble 3 4h: Data nibble 4 5h: Data nibble 5 6h: Data nibble 6 7h: Data nibble 7 8h: Data nibble 8 9h: Reserved Ah: CRC nibble 1 Bh: CRC nibble 2 Ch-Eh: Reserved Fh: Status
3-0	D1_NIB0	R/W	0h	Data1 nibble 0 sorting format register. This field configures where the received nibbles are stored in the data1 map for FIFO or direct map mode. D1_NIB0[3:0] 0h: None 1h: Data nibble 1 2h: Data nibble 2 3h: Data nibble 3 4h: Data nibble 4 5h: Data nibble 5 6h: Data nibble 6 7h: Data nibble 7 8h: Data nibble 8 9h: Reserved Ah: CRC nibble 1 Bh: CRC nibble 2 Ch-Eh: Reserved Fh: Status

42.8.2.13 CSENT_TO Register (Offset = 5Ch) [Reset = 00000000h]

CSENT_TO is shown in Figure 42-25 and described in Table 42-24.

Return to the Summary Table.

CSENT's Timeout register

Figure 42-25. CSENT_TO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO_VAL																															
R/W-0h																															

Table 42-24. CSENT_TO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TO_VAL	R/W	0h	TO_VAL is used to specify the timeout setting to indicate that no synchronization pulse has been received within the timeout time specified. This can be used for functional safety and a timeout interrupt will occur when this time as expired. The Timeout functionality is enabled only after receiving a synchronization pulse. The value is specified in clock tick units, and it can be disabled by setting the value to 0, which is the default value.

42.8.2.14 CSENT_RXD Register (Offset = 60h) [Reset = 00000001h]

CSENT_RXD is shown in Figure 42-26 and described in Table 42-25.

Return to the Summary Table.

CSENT Receiver Serial Data input register

Figure 42-26. CSENT_RXD Register

31	30	29	28	27	26	25	24
RSVD							
R-0h							
23	22	21	20	19	18	17	16
RSVD							
R-0h							
15	14	13	12	11	10	9	8
RSVD							
R-0h							
7	6	5	4	3	2	1	0
RSVD							RXD_I_R
R-0h							R-1h

Table 42-25. CSENT_RXD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RSVD	R	0h	Reserved bits
0	RXD_I_R	R	1h	RXD_I_R contains CSENT synchronized serial data input value. This can be used for debugging purposes.

42.8.2.15 RXVAL_CNT Register (Offset = 64h) [Reset = 00000000h]

RXVAL_CNT is shown in Figure 42-27 and described in Table 42-26.

Return to the Summary Table.

CSENT Rx Valid Frames Counter

Figure 42-27. RXVAL_CNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								RXVALID_CNT_R							
R-0h								R-0h							

Table 42-26. RXVAL_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RSVD	R	0h	Reserved bits
7-0	RXVALID_CNT_R	R	0h	RXVALID_CNT_R contains the number of the valid CSENT frames received. The maximum value of this counter is 255. It can be reset to 0 via SWR_RXVAL_CNT Software Reset or CSENT Software Reset or System Reset. This can be used for debugging purposes.

42.8.2.16 RXDEDGE_CNT Register (Offset = 68h) [Reset = 00000000h]

RXDEDGE_CNT is shown in Figure 42-28 and described in Table 42-27.

Return to the Summary Table.

CSENT Receiver Serial Data Input Falling-edge Counter

Figure 42-28. RXDEDGE_CNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								RXDEDGE_CNT_R							
R-0h								R-0h							

Table 42-27. RXDEDGE_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RSVD	R	0h	Reserved bits
7-0	RXDEDGE_CNT_R	R	0h	RXDEDGE_CNT_R contains the number of CSENT synchronized falling-edges of the Receiver Serial Data input. The maximum value of this counter is 255. It can be reset to 0 via SWR_RXDEDGE_CNT Software Reset or CSENT Software Reset or System Reset. This can be used for debugging purposes.

42.8.2.17 SWR_RXVAL_CNT Register (Offset = 6Ch) [Reset = 00000000h]

SWR_RXVAL_CNT is shown in Figure 42-29 and described in Table 42-28.

Return to the Summary Table.

Valid CSENT frames counter Sofware Reset

Figure 42-29. SWR_RXVAL_CNT Register

31	30	29	28	27	26	25	24
RSVD							
W-0h							
23	22	21	20	19	18	17	16
RSVD							
W-0h							
15	14	13	12	11	10	9	8
RSVD							
W-0h							
7	6	5	4	3	2	1	0
RSVD							SWR_RXVAL_CNT
W-0h							W-0h

Table 42-28. SWR_RXVAL_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RSVD	W	0h	Reserved bits
0	SWR_RXVAL_CNT	W	0h	0h: No software reset on RXVAL_CNT 1h: It will cause software reset on the RXVAL_CNT register to 0h. This can be used for debugging purposes

42.8.2.18 SWR_RXDEDGE_CNT Register (Offset = 70h) [Reset = 00000000h]

SWR_RXDEDGE_CNT is shown in Figure 42-30 and described in Table 42-29.

Return to the Summary Table.

CSENT Receiver Serial Data input Falling-edge Counter Software Reset

Figure 42-30. SWR_RXDEDGE_CNT Register

31	30	29	28	27	26	25	24
RSVD							
W-0h							
23	22	21	20	19	18	17	16
RSVD							
W-0h							
15	14	13	12	11	10	9	8
RSVD							
W-0h							
7	6	5	4	3	2	1	0
RSVD							SWR_RXDEDGE_CNT
W-0h							W-0h

Table 42-29. SWR_RXDEDGE_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RSVD	W	0h	Reserved bits
0	SWR_RXDEDGE_CNT	W	0h	0h: No Software reset on the RXDEDGE_CNT 1h: It will cause software reset on the RXDEDGE_CNT register to 0h. This can be used for debugging purposes

42.8.2.19 CSENT_VERSION Register (Offset = 7Ch) [Reset = 01000001h]

CSENT_VERSION is shown in Figure 42-31 and described in Table 42-30.

Return to the Summary Table.

CSENT_XRF Version Number register

Figure 42-31. CSENT_VERSION Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VERSION																															
R-01000001h																															

Table 42-30. CSENT_VERSION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VERSION	R	01000001h	VERSION can programmatically provide the version number of the CSENT_XRF IP core.

42.8.3 SENT_MEM Registers

Table 42-31 lists the memory-mapped registers for the SENT_MEM registers. All register offset addresses not listed in Table 42-31 should be considered as reserved locations and the register contents should not be modified.

Table 42-31. SENT_MEM Registers

Offset	Acronym	Register Name
400h + formula	MDATA_y	Data word 'N' in TPRAM memory. The SENT TPRAM contains fast channel SENT frames whether in FIFO or Direct Map mode. In Direct Map mode, the user must read the SENT frame data out from this memory (refer to the section regarding Receive Modes for more details)

Complex bit access types are encoded to fit into small table cells. Table 42-32 shows the codes that are used for access types in this section.

Table 42-32. SENT_MEM Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

42.8.3.1 MDATA_y Register (Offset = 400h + formula) [Reset = 00000000h]

MDATA_y is shown in Figure 42-32 and described in Table 42-33.

Return to the Summary Table.

32-bit data word 'N' in TPRAM memory. The SENT TPRAM contains fast channel SENT frames whether in FIFO or Direct Map mode. In Direct Map mode, the user must read the SENT frame data out from this memory (refer to the section regarding Receive Modes for more details)

Offset = 400h + (y * 4h); where y = 0h to Fh

Figure 42-32. MDATA_y Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT																															
R-0h																															

Table 42-33. MDATA_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DAT	R	0h	32-bit data word

42.8.4 SENT_MTPG Registers

Table 42-34 lists the memory-mapped registers for the SENT_MTPG registers. All register offset addresses not listed in Table 42-34 should be considered as reserved locations and the register contents should not be modified.

Table 42-34. SENT_MTPG Registers

Offset	Acronym	Register Name
800h	BC_MTP_EN	Broadcast Channel MTP Enable Register
804h	BC_MTP_CMP1	Broadcast Channel MTP Compare1 Register
808h	BC_MTP_CMP2	Broadcast Channel MTP Compare2 Register
80Ch	BC_MTP_CMP3	Broadcast Channel MTP Compare3 Register
810h	BC_MTP_CMP4	Broadcast Channel MTP Compare4 Register
814h	BC_MTP_CMP5	Broadcast Channel MTP Compare5 Register
818h	BC_MTP_CMP6	Broadcast Channel MTP Compare6 Register
81Ch	BC_MTP_CMP7	Broadcast Channel MTP Compare7 Register
820h	BC_MTP_CMP8	Broadcast Channel MTP Compare8 Register
824h	BC_MTP_CMP9	Broadcast Channel MTP Compare9 Register
82Ch	BC_MTP_PERIOD	Broadcast Channel MTP Period Register.
834h	BC_TRIGSEL	Broadcast Channel MTP Trigger Select Register
838h	BC_MTP_SWTR	Broadcast Channel MTP Software Trigger Register
900h	S1_MTP_EN	Sensor 1 Channel MTP Enable Register
904h	S1_MTP_CMP1	Sensor 1 Channel MTP Compare1 Register
908h	S1_MTP_CMP2	Sensor 1 Channel MTP Compare2 Register
90Ch	S1_MTP_CMP3	Sensor 1 Channel MTP Compare3 Register
910h	S1_MTP_CMP4	Sensor 1 Channel MTP Compare4 Register
914h	S1_MTP_CMP5	Sensor 1 Channel MTP Compare5 Register
918h	S1_MTP_CMP6	Sensor 1 Channel MTP Compare6 Register
91Ch	S1_MTP_CMP7	Sensor 1 Channel MTP Compare7 Register
920h	S1_MTP_CMP8	Sensor 1 Channel MTP Compare8 Register
924h	S1_MTP_CMP9	Sensor 1 Channel MTP Compare9 Register
928h	S1_MTP_CMP10RE	Sensor 1 Channel MTP Compare10 Receiver Enable Register
92Ch	S1_MTP_PERIOD	Sensor 1 Channel MTP Period Register
930h	S1_MTP_TO	Sensor 1 Channel MTP TimeOut Register
934h	S1_TRIGSEL	Sensor 1 Channel MTP Trigger Select Register
938h	S1_MTP_SWTR	Sensor 1 Channel MTP Software Trigger Register
A00h	S2_MTP_EN	Sensor 2 Channel MTP Enable Register
A04h	S2_MTP_CMP1	Sensor 2 Channel MTP Compare1 Register
A08h	S2_MTP_CMP2	Sensor 2 Channel MTP Compare2 Register
A0Ch	S2_MTP_CMP3	Sensor 2 Channel MTP Compare3 Register
A10h	S2_MTP_CMP4	Sensor 2 Channel MTP Compare4 Register
A14h	S2_MTP_CMP5	Sensor 2 Channel MTP Compare5 Register
A18h	S2_MTP_CMP6	Sensor 2 Channel MTP Compare6 Register
A1Ch	S2_MTP_CMP7	Sensor 2 Channel MTP Compare7 Register
A20h	S2_MTP_CMP8	Sensor 2 Channel MTP Compare8 Register
A24h	S2_MTP_CMP9	Sensor 2 Channel MTP Compare9 Register
A28h	S2_MTP_CMP10RE	Sensor 2 Channel MTP Compare10 Receiver Enable Register
A2Ch	S2_MTP_PERIOD	Sensor 2 Channel MTP Period Register

Table 42-34. SENT_MTPG Registers (continued)

Offset	Acronym	Register Name
A30h	S2_MTP_TO	Sensor 2 Channel MTP TimeOut Register
A34h	S2_TRIGSEL	Sensor 2 Channel MTP Trigger Select Register
A38h	S2_MTP_SWTR	Sensor 2 Channel MTP Software Trigger Register
B00h	S3_MTP_EN	Sensor 3 Channel MTP Enable Register
B04h	S3_MTP_CMP1	Sensor 3 Channel MTP Compare1 Register
B08h	S3_MTP_CMP2	Sensor 3 Channel MTP Compare2 Register
B0Ch	S3_MTP_CMP3	Sensor 3 Channel MTP Compare3 Register
B10h	S3_MTP_CMP4	Sensor 3 Channel MTP Compare4 Register
B14h	S3_MTP_CMP5	Sensor 3 Channel MTP Compare5 Register
B18h	S3_MTP_CMP6	Sensor 3 Channel MTP Compare6 Register
B1Ch	S3_MTP_CMP7	Sensor 3 Channel MTP Compare7 Register
B20h	S3_MTP_CMP8	Sensor 3 Channel MTP Compare8 Register
B24h	S3_MTP_CMP9	Sensor 3 Channel MTP Compare9 Register
B28h	S3_MTP_CMP10RE	Sensor 3 Channel MTP Compare10 Receiver Enable Register
B2Ch	S3_MTP_PERIOD	Sensor 3 Channel MTP Period Register
B30h	S3_MTP_TO	Sensor 3 Channel MTP TimeOut Register
B34h	S3_TRIGSEL	Sensor 3 Channel MTP Trigger Select Register
B38h	S3_MTP_SWTR	Sensor 3 Channel MTP Software Trigger Register
C00h	S4_MTP_EN	Sensor 4 Channel MTP Enable Register
C04h	S4_MTP_CMP1	Sensor 4 Channel MTP Compare1 Register
C08h	S4_MTP_CMP2	Sensor 4 Channel MTP Compare2 Register
C0Ch	S4_MTP_CMP3	Sensor 4 Channel MTP Compare3 Register
C10h	S4_MTP_CMP4	Sensor 4 Channel MTP Compare4 Register
C14h	S4_MTP_CMP5	Sensor 4 Channel MTP Compare5 Register
C18h	S4_MTP_CMP6	Sensor 4 Channel MTP Compare6 Register
C1Ch	S4_MTP_CMP7	Sensor 4 Channel MTP Compare7 Register
C20h	S4_MTP_CMP8	Sensor 4 Channel MTP Compare8 Register
C24h	S4_MTP_CMP9	Sensor 4 Channel MTP Compare9 Register
C28h	S4_MTP_CMP10RE	Sensor 4 Channel MTP Compare10 Receiver Enable Register
C2Ch	S4_MTP_PERIOD	Sensor 4 Channel MTP Period Register
C30h	S4_MTP_TO	Sensor 4 Channel MTP TimeOut Register
C34h	S4_TRIGSEL	Sensor 4 Channel MTP Trigger Select Register
C38h	S4_MTP_SWTR	Sensor 4 Channel MTP Software Trigger Register
F00h	WAITTIME	Global Waittime Register
F04h	TPGENSTAT	Master Trigger Pulse Global Status Register
F30h	MTP_VERSION	MTPG IP Core Version Number Register
F3Ch	MTP_SWR	MTPG Global Software Reset Register. Writing 1 to this register will cause Software Reset

Complex bit access types are encoded to fit into small table cells. Table 42-35 shows the codes that are used for access types in this section.

Table 42-35. SENT_MTPG Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read

Table 42-35. SENT_MTPG Access Type Codes (continued)

Access Type	Code	Description
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

42.8.4.1 BC_MTP_EN Register (Offset = 800h) [Reset = 00000000h]

BC_MTP_EN is shown in Figure 42-33 and described in Table 42-36.

Return to the Summary Table.

Broadcast Channel MTP Enable Register

Figure 42-33. BC_MTP_EN Register

31	30	29	28	27	26	25	24
RSVD							
R-0h							
23	22	21	20	19	18	17	16
RSVD							
R-0h							
15	14	13	12	11	10	9	8
RSVD							
R-0h							
7	6	5	4	3	2	1	0
RSVD							MTP_EN
R-0h							R/W-0h

Table 42-36. BC_MTP_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RSVD	R	0h	Reserved bits
0	MTP_EN	R/W	0h	0h: No trigger pulses generated 1h: The Master Trigger Pulse Generator of the Broadcast Channel will generate trigger pulses when it is triggered via Hardware or Software

42.8.4.2 BC_MTP_CMP1 Register (Offset = 804h) [Reset = 00000000h]

BC_MTP_CMP1 is shown in Figure 42-34 and described in Table 42-37.

Return to the Summary Table.

Broadcast Channel MTP Compare1 Register

Figure 42-34. BC_MTP_CMP1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-37. BC_MTP_CMP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the BC_MTP_CMP1 register contains the time (in clock tick unit) of the Broadcast Channel's Master Trigger Pulse Generator output to be toggled. No toggle if TOGGLETIME = 0.

42.8.4.3 BC_MTP_CMP2 Register (Offset = 808h) [Reset = 00000000h]

BC_MTP_CMP2 is shown in Figure 42-35 and described in Table 42-38.

Return to the Summary Table.

Broadcast Channel MTP Compare2 Register

Figure 42-35. BC_MTP_CMP2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-38. BC_MTP_CMP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the BC_MTP_CMP2 register contains the time (in clock tick unit) of the Broadcast Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than BC_MTP_CMP1's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.4 BC_MTP_CMP3 Register (Offset = 80Ch) [Reset = 00000000h]

BC_MTP_CMP3 is shown in Figure 42-36 and described in Table 42-39.

Return to the Summary Table.

Broadcast Channel MTP Compare3 Register

Figure 42-36. BC_MTP_CMP3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-39. BC_MTP_CMP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the BC_MTP_CMP3 register contains the time (in clock tick unit) of the Broadcast Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than BC_MTP_CMP2's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.5 BC_MTP_CMP4 Register (Offset = 810h) [Reset = 00000000h]

BC_MTP_CMP4 is shown in Figure 42-37 and described in Table 42-40.

Return to the Summary Table.

Broadcast Channel MTP Compare4 Register

Figure 42-37. BC_MTP_CMP4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-40. BC_MTP_CMP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the BC_MTP_CMP4 register contains the time (in clock tick unit) of the Broadcast Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than BC_MTP_CMP3's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.6 BC_MTP_CMP5 Register (Offset = 814h) [Reset = 00000000h]

BC_MTP_CMP5 is shown in Figure 42-38 and described in Table 42-41.

Return to the Summary Table.

Broadcast Channel MTP Compare5 Register

Figure 42-38. BC_MTP_CMP5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-41. BC_MTP_CMP5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the BC_MTP_CMP5 register contains the time (in clock tick unit) of the Broadcast Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than BC_MTP_CMP4's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.7 BC_MTP_CMP6 Register (Offset = 818h) [Reset = 00000000h]

BC_MTP_CMP6 is shown in Figure 42-39 and described in Table 42-42.

Return to the Summary Table.

Broadcast Channel MTP Compare6 Register

Figure 42-39. BC_MTP_CMP6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-42. BC_MTP_CMP6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the BC_MTP_CMP6 register contains the time (in clock tick unit) of the Broadcast Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than BC_MTP_CMP5's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.8 BC_MTP_CMP7 Register (Offset = 81Ch) [Reset = 00000000h]

BC_MTP_CMP7 is shown in Figure 42-40 and described in Table 42-43.

Return to the Summary Table.

Broadcast Channel MTP Compare7 Register

Figure 42-40. BC_MTP_CMP7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-43. BC_MTP_CMP7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the BC_MTP_CMP7 register contains the time (in clock tick unit) of the Broadcast Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than BC_MTP_CMP6's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.9 BC_MTP_CMP8 Register (Offset = 820h) [Reset = 00000000h]

BC_MTP_CMP8 is shown in Figure 42-41 and described in Table 42-44.

Return to the Summary Table.

Broadcast Channel MTP Compare8 Register

Figure 42-41. BC_MTP_CMP8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-44. BC_MTP_CMP8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the BC_MTP_CMP8 register contains the time (in clock tick unit) of the Broadcast Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than BC_MTP_CMP7's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.10 BC_MTP_CMP9 Register (Offset = 824h) [Reset = 00000000h]

BC_MTP_CMP9 is shown in Figure 42-42 and described in Table 42-45.

Return to the Summary Table.

Broadcast Channel MTP Compare9 Register

Figure 42-42. BC_MTP_CMP9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-45. BC_MTP_CMP9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the BC_MTP_CMP9 register contains the time (in clock tick unit) of the Broadcast Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than BC_MTP_CMP8's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.11 BC_MTP_PERIOD Register (Offset = 82Ch) [Reset = 00000000h]

BC_MTP_PERIOD is shown in Figure 42-43 and described in Table 42-46.

Return to the Summary Table.

Broadcast Channel MTP Period Register.

Figure 42-43. BC_MTP_PERIOD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					PERIOD										
R-0h																					R/W-0h										

Table 42-46. BC_MTP_PERIOD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	PERIOD	R/W	0h	PERIOD of the BC_MTP_PERIOD register indicates the total time that the master trigger pulse output waveform is active. The PERIOD must be larger than all Broadcast Channel Compares' TOGGLETIME value. The value is specified in clock tick units.

42.8.4.12 BC_TRIGSEL Register (Offset = 834h) [Reset = 00000000h]

BC_TRIGSEL is shown in Figure 42-44 and described in Table 42-47.

Return to the Summary Table.

Broadcast Channel MTP Trigger Select Register

Figure 42-44. BC_TRIGSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																									TRIGSEL						
R-0h																									R/W-0h						

Table 42-47. BC_TRIGSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RSVD	R	0h	Reserved bits
6-0	TRIGSEL	R/W	0h	Specifies which hardware trigger source enables the Broadcast Channel Master Trigger Pulse Generator. The possible internal triggers are: 0h: Disable (no hardware triggers) 1h: Broadcast Trigger done 2h: Sensor 1 Trigger done 3h: Sensor 2 Trigger done 4h: Sensor 3 Trigger done 5h: Sensor 4 Trigger done External triggers are also available and are described by the 'Channel Triggers' section.

42.8.4.13 BC_MTP_SWTR Register (Offset = 838h) [Reset = 00000000h]

BC_MTP_SWTR is shown in Figure 42-45 and described in Table 42-48.

Return to the Summary Table.

Broadcast Channel MTP Software Trigger Register

Figure 42-45. BC_MTP_SWTR Register

31	30	29	28	27	26	25	24
RSVD							
W-0h							
23	22	21	20	19	18	17	16
RSVD							
W-0h							
15	14	13	12	11	10	9	8
RSVD							
W-0h							
7	6	5	4	3	2	1	0
RSVD							SWTR
W-0h							W-0h

Table 42-48. BC_MTP_SWTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RSVD	W	0h	Reserved bits
0	SWTR	W	0h	0h: No software trigger request 1h: Broadcast Channel's software trigger request. The SWTR bit will be cleared automatically after the Broadcast software trigger request is in service. The BC_MTP_EN must active for SWTR to function

42.8.4.14 S1_MTP_EN Register (Offset = 900h) [Reset = 00000000h]

S1_MTP_EN is shown in Figure 42-46 and described in Table 42-49.

Return to the Summary Table.

Sensor 1 Channel MTP Enable Register

Figure 42-46. S1_MTP_EN Register

31	30	29	28	27	26	25	24
RSVD							
R-0h							
23	22	21	20	19	18	17	16
RSVD							
R-0h							
15	14	13	12	11	10	9	8
RSVD							
R-0h							
7	6	5	4	3	2	1	0
RSVD							MTP_EN
R-0h							R/W-0h

Table 42-49. S1_MTP_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RSVD	R	0h	Reserved bits
0	MTP_EN	R/W	0h	0h: No trigger pulses generated 1h: The Master Trigger Pulse Generator of the Sensor 1 Channel will generate trigger pulses when it is triggered via Hardware or Software

42.8.4.15 S1_MTP_CMP1 Register (Offset = 904h) [Reset = 00000000h]

S1_MTP_CMP1 is shown in Figure 42-47 and described in Table 42-50.

Return to the Summary Table.

Sensor 1 Channel MTP Compare1 Register

Figure 42-47. S1_MTP_CMP1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-50. S1_MTP_CMP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S1_MTP_CMP1 register contains the time (in clock tick unit) of the Sensor 1 Channel's Master Trigger Pulse Generator output to be toggled. No toggle if TOGGLETIME = 0.

42.8.4.16 S1_MTP_CMP2 Register (Offset = 908h) [Reset = 00000000h]

S1_MTP_CMP2 is shown in Figure 42-48 and described in Table 42-51.

Return to the Summary Table.

Sensor 1 Channel MTP Compare2 Register

Figure 42-48. S1_MTP_CMP2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-51. S1_MTP_CMP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S1_MTP_CMP2 register contains the time (in clock tick unit) of the Sensor 1 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S1_MTP_CMP1's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.17 S1_MTP_CMP3 Register (Offset = 90Ch) [Reset = 00000000h]

S1_MTP_CMP3 is shown in Figure 42-49 and described in Table 42-52.

Return to the Summary Table.

Sensor 1 Channel MTP Compare3 Register

Figure 42-49. S1_MTP_CMP3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-52. S1_MTP_CMP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S1_MTP_CMP3 register contains the time (in clock tick unit) of the Sensor 1 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S1_MTP_CMP2's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.18 S1_MTP_CMP4 Register (Offset = 910h) [Reset = 00000000h]

S1_MTP_CMP4 is shown in Figure 42-50 and described in Table 42-53.

Return to the Summary Table.

Sensor 1 Channel MTP Compare4 Register

Figure 42-50. S1_MTP_CMP4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-53. S1_MTP_CMP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S1_MTP_CMP4 register contains the time (in clock tick unit) of the Sensor 1 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S1_MTP_CMP3's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.19 S1_MTP_CMP5 Register (Offset = 914h) [Reset = 00000000h]

S1_MTP_CMP5 is shown in Figure 42-51 and described in Table 42-54.

Return to the Summary Table.

Sensor 1 Channel MTP Compare5 Register

Figure 42-51. S1_MTP_CMP5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-54. S1_MTP_CMP5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S1_MTP_CMP5 register contains the time (in clock tick unit) of the Sensor 1 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S1_MTP_CMP4's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.20 S1_MTP_CMP6 Register (Offset = 918h) [Reset = 00000000h]

S1_MTP_CMP6 is shown in Figure 42-52 and described in Table 42-55.

Return to the Summary Table.

Sensor 1 Channel MTP Compare6 Register

Figure 42-52. S1_MTP_CMP6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-55. S1_MTP_CMP6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S1_MTP_CMP6 register contains the time (in clock tick unit) of the Sensor 1 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S1_MTP_CMP5's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.21 S1_MTP_CMP7 Register (Offset = 91Ch) [Reset = 00000000h]

S1_MTP_CMP7 is shown in Figure 42-53 and described in Table 42-56.

Return to the Summary Table.

Sensor 1 Channel MTP Compare7 Register

Figure 42-53. S1_MTP_CMP7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-56. S1_MTP_CMP7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S1_MTP_CMP7 register contains the time (in clock tick unit) of the Sensor 1 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S1_MTP_CMP6's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.22 S1_MTP_CMP8 Register (Offset = 920h) [Reset = 00000000h]

S1_MTP_CMP8 is shown in Figure 42-54 and described in Table 42-57.

Return to the Summary Table.

Sensor 1 Channel MTP Compare8 Register

Figure 42-54. S1_MTP_CMP8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-57. S1_MTP_CMP8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S1_MTP_CMP8 register contains the time (in clock tick unit) of the Sensor 1 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S1_MTP_CMP7's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.23 S1_MTP_CMP9 Register (Offset = 924h) [Reset = 00000000h]

S1_MTP_CMP9 is shown in Figure 42-55 and described in Table 42-58.

Return to the Summary Table.

Sensor 1 Channel MTP Compare9 Register

Figure 42-55. S1_MTP_CMP9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-58. S1_MTP_CMP9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S1_MTP_CMP9 register contains the time (in clock tick unit) of the Sensor 1 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S1_MTP_CMP8's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.24 S1_MTP_CMP10RE Register (Offset = 928h) [Reset = 00000000h]

S1_MTP_CMP10RE is shown in Figure 42-56 and described in Table 42-59.

Return to the Summary Table.

Sensor 1 Channel MTP Compare10 Receiver Enable Register

Figure 42-56. S1_MTP_CMP10RE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-59. S1_MTP_CMP10RE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S1_MTP_CMP10RE defines when to enable the CSENT Receiver while the Sensor 1's trigger pulses are generated. This TOGGLETIME will be compared independently, since it might be higher or lower than other Sensor 1's compare register value. The value is specified in clock tick units.

42.8.4.25 S1_MTP_PERIOD Register (Offset = 92Ch) [Reset = 00000000h]

S1_MTP_PERIOD is shown in Figure 42-57 and described in Table 42-60.

Return to the Summary Table.

Sensor 1 Channel MTP Period Register

Figure 42-57. S1_MTP_PERIOD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					PERIOD										
R-0h																					R/W-0h										

Table 42-60. S1_MTP_PERIOD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	PERIOD	R/W	0h	PERIOD of the S1_MTP_PERIOD register indicates the total time that the master trigger pulse output waveform is active. The PERIOD must be larger than all Sensor 1 Channel Compares' TOGGLETIME value. The value is specified in clock tick units.

42.8.4.26 S1_MTP_TO Register (Offset = 930h) [Reset = 00000000h]

S1_MTP_TO is shown in Figure 42-58 and described in Table 42-61.

Return to the Summary Table.

Sensor 1 Channel MTP TimeOut Register

Figure 42-58. S1_MTP_TO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEOUT																															
R/W-0h																															

Table 42-61. S1_MTP_TO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TIMEOUT	R/W	0h	The TIMEOUT of the S1_MTP_TO register indicates a time to initiate an interrupt if the CSENT Receiver has not received response (low pulse of calibration/synchronization) from Sensor 1 within timeout time after the Master trigger pulse is generated. The value is specified in clock tick units.

42.8.4.27 S1_TRIGSEL Register (Offset = 934h) [Reset = 00000000h]

S1_TRIGSEL is shown in Figure 42-59 and described in Table 42-62.

Return to the Summary Table.

Sensor 1 Channel MTP Trigger Select Register

Figure 42-59. S1_TRIGSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																									TRIGSEL						
R-0h																									R/W-0h						

Table 42-62. S1_TRIGSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RSVD	R	0h	Reserved bits
6-0	TRIGSEL	R/W	0h	Specifies which hardware trigger source enables the Sensor 1 Channel Master Trigger Pulse Generator. The possible internal triggers are: 0h: Disable (no hardware triggers) 1h: Broadcast Trigger done 2h: Sensor 1 Trigger done 3h: Sensor 2 Trigger done 4h: Sensor 3 Trigger done 5h: Sensor 4 Trigger done External triggers are also available and are described by the 'Channel Triggers' section.

42.8.4.28 S1_MTP_SWTR Register (Offset = 938h) [Reset = 00000000h]

S1_MTP_SWTR is shown in Figure 42-60 and described in Table 42-63.

Return to the Summary Table.

Sensor 1 Channel MTP Software Trigger Register

Figure 42-60. S1_MTP_SWTR Register

31	30	29	28	27	26	25	24
RSVD							
W-0h							
23	22	21	20	19	18	17	16
RSVD							
W-0h							
15	14	13	12	11	10	9	8
RSVD							
W-0h							
7	6	5	4	3	2	1	0
RSVD							SWTR
W-0h							W-0h

Table 42-63. S1_MTP_SWTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RSVD	W	0h	Reserved bits
0	SWTR	W	0h	0h: No software trigger request 1h: Sensor 1 Channel's software trigger request. The SWTR bit will be cleared automatically after the Sensor 1 software trigger request is in service. The S1_MTP_EN must active for SWTR to function

42.8.4.29 S2_MTP_EN Register (Offset = A00h) [Reset = 00000000h]

S2_MTP_EN is shown in Figure 42-61 and described in Table 42-64.

Return to the Summary Table.

Sensor 2 Channel MTP Enable Register

Figure 42-61. S2_MTP_EN Register

31	30	29	28	27	26	25	24
RSVD							
R-0h							
23	22	21	20	19	18	17	16
RSVD							
R-0h							
15	14	13	12	11	10	9	8
RSVD							
R-0h							
7	6	5	4	3	2	1	0
RSVD							MTP_EN
R-0h							R/W-0h

Table 42-64. S2_MTP_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RSVD	R	0h	Reserved bits
0	MTP_EN	R/W	0h	0h: No trigger pulses generated 1h: The Master Trigger Pulse Generator of the Sensor 2 Channel will generate trigger pulses when it is triggered via Hardware or Software

42.8.4.30 S2_MTP_CMP1 Register (Offset = A04h) [Reset = 00000000h]

S2_MTP_CMP1 is shown in Figure 42-62 and described in Table 42-65.

Return to the Summary Table.

Sensor 2 Channel MTP Compare1 Register

Figure 42-62. S2_MTP_CMP1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-65. S2_MTP_CMP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S2_MTP_CMP1 register contains the time (in clock tick unit) of the Sensor 2 Channel's Master Trigger Pulse Generator output to be toggled. No toggle if TOGGLETIME = 0.

42.8.4.31 S2_MTP_CMP2 Register (Offset = A08h) [Reset = 00000000h]

S2_MTP_CMP2 is shown in Figure 42-63 and described in Table 42-66.

Return to the Summary Table.

Sensor 2 Channel MTP Compare2 Register

Figure 42-63. S2_MTP_CMP2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-66. S2_MTP_CMP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S2_MTP_CMP2 register contains the time (in clock tick unit) of the Sensor 2 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S2_MTP_CMP1's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.32 S2_MTP_CMP3 Register (Offset = A0Ch) [Reset = 00000000h]

S2_MTP_CMP3 is shown in Figure 42-64 and described in Table 42-67.

Return to the Summary Table.

Sensor 2 Channel MTP Compare3 Register

Figure 42-64. S2_MTP_CMP3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-67. S2_MTP_CMP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S2_MTP_CMP3 register contains the time (in clock tick unit) of the Sensor 2 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S2_MTP_CMP2's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.33 S2_MTP_CMP4 Register (Offset = A10h) [Reset = 00000000h]

S2_MTP_CMP4 is shown in Figure 42-65 and described in Table 42-68.

Return to the Summary Table.

Sensor 2 Channel MTP Compare4 Register

Figure 42-65. S2_MTP_CMP4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-68. S2_MTP_CMP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S2_MTP_CMP4 register contains the time (in clock tick unit) of the Sensor 2 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S2_MTP_CMP3's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.34 S2_MTP_CMP5 Register (Offset = A14h) [Reset = 00000000h]

S2_MTP_CMP5 is shown in Figure 42-66 and described in Table 42-69.

Return to the Summary Table.

Sensor 2 Channel MTP Compare5 Register

Figure 42-66. S2_MTP_CMP5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-69. S2_MTP_CMP5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S2_MTP_CMP5 register contains the time (in clock tick unit) of the Sensor 2 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S2_MTP_CMP4's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.35 S2_MTP_CMP6 Register (Offset = A18h) [Reset = 00000000h]

S2_MTP_CMP6 is shown in Figure 42-67 and described in Table 42-70.

Return to the Summary Table.

Sensor 2 Channel MTP Compare6 Register

Figure 42-67. S2_MTP_CMP6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-70. S2_MTP_CMP6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S2_MTP_CMP6 register contains the time (in clock tick unit) of the Sensor 2 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S2_MTP_CMP5's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.36 S2_MTP_CMP7 Register (Offset = A1Ch) [Reset = 00000000h]

S2_MTP_CMP7 is shown in Figure 42-68 and described in Table 42-71.

Return to the Summary Table.

Sensor 2 Channel MTP Compare7 Register

Figure 42-68. S2_MTP_CMP7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-71. S2_MTP_CMP7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S2_MTP_CMP7 register contains the time (in clock tick unit) of the Sensor 2 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S2_MTP_CMP6's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.37 S2_MTP_CMP8 Register (Offset = A20h) [Reset = 00000000h]

S2_MTP_CMP8 is shown in Figure 42-69 and described in Table 42-72.

Return to the Summary Table.

Sensor 2 Channel MTP Compare8 Register

Figure 42-69. S2_MTP_CMP8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-72. S2_MTP_CMP8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S2_MTP_CMP8 register contains the time (in clock tick unit) of the Sensor 2 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S2_MTP_CMP7's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.38 S2_MTP_CMP9 Register (Offset = A24h) [Reset = 00000000h]

S2_MTP_CMP9 is shown in Figure 42-70 and described in Table 42-73.

Return to the Summary Table.

Sensor 2 Channel MTP Compare9 Register

Figure 42-70. S2_MTP_CMP9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-73. S2_MTP_CMP9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S2_MTP_CMP9 register contains the time (in clock tick unit) of the Sensor 2 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S2_MTP_CMP8's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.39 S2_MTP_CMP10RE Register (Offset = A28h) [Reset = 00000000h]

S2_MTP_CMP10RE is shown in Figure 42-71 and described in Table 42-74.

Return to the Summary Table.

Sensor 2 Channel MTP Compare10 Receiver Enable Register

Figure 42-71. S2_MTP_CMP10RE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-74. S2_MTP_CMP10RE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S2_MTP_CMP10RE defines when to enable the CSENT Receiver while the Sensor 2's trigger pulses are generated. This TOGGLETIME will be compared independently, since it might be higher or lower than other Sensor 2's compare register value. The value is specified in clock tick units.

42.8.4.40 S2_MTP_PERIOD Register (Offset = A2Ch) [Reset = 00000000h]

S2_MTP_PERIOD is shown in Figure 42-72 and described in Table 42-75.

Return to the Summary Table.

Sensor 2 Channel MTP Period Register

Figure 42-72. S2_MTP_PERIOD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					PERIOD										
R-0h																					R/W-0h										

Table 42-75. S2_MTP_PERIOD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	PERIOD	R/W	0h	PERIOD of the S2_MTP_PERIOD register indicates the total time that the master trigger pulse output waveform is active. The PERIOD must be larger than all Sensor 2 Channel Compares' TOGGLETIME value. The value is specified in clock tick units.

42.8.4.41 S2_MTP_TO Register (Offset = A30h) [Reset = 00000000h]

S2_MTP_TO is shown in Figure 42-73 and described in Table 42-76.

Return to the Summary Table.

Sensor 2 Channel MTP TimeOut Register

Figure 42-73. S2_MTP_TO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEOUT																															
R/W-0h																															

Table 42-76. S2_MTP_TO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TIMEOUT	R/W	0h	The TIMEOUT of the S2_MTP_TO register indicates a time to initiate an interrupt if the CSENT Receiver has not received response (low pulse of calibration/synchronization) from Sensor 2 within timeout time after master trigger pulse is generated. The value is specified in clock tick units.

42.8.4.42 S2_TRIGSEL Register (Offset = A34h) [Reset = 00000000h]

S2_TRIGSEL is shown in Figure 42-74 and described in Table 42-77.

Return to the Summary Table.

Sensor 2 Channel MTP Trigger Select Register

Figure 42-74. S2_TRIGSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																									TRIGSEL						
R-0h																									R/W-0h						

Table 42-77. S2_TRIGSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RSVD	R	0h	Reserved bits
6-0	TRIGSEL	R/W	0h	Specifies which hardware trigger source enables the Sensor 2 Channel Master Trigger Pulse Generator. The possible internal triggers are: 0h: Disable (no hardware triggers) 1h: Broadcast Trigger done 2h: Sensor 1 Trigger done 3h: Sensor 2 Trigger done 4h: Sensor 3 Trigger done 5h: Sensor 4 Trigger done External triggers are also available and are described by the 'Channel Triggers' section.

42.8.4.43 S2_MTP_SWTR Register (Offset = A38h) [Reset = 00000000h]

S2_MTP_SWTR is shown in Figure 42-75 and described in Table 42-78.

Return to the Summary Table.

Sensor 2 Channel MTP Software Trigger Register

Figure 42-75. S2_MTP_SWTR Register

31	30	29	28	27	26	25	24
RSVD							
W-0h							
23	22	21	20	19	18	17	16
RSVD							
W-0h							
15	14	13	12	11	10	9	8
RSVD							
W-0h							
7	6	5	4	3	2	1	0
RSVD							SWTR
W-0h							W-0h

Table 42-78. S2_MTP_SWTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RSVD	W	0h	Reserved bits
0	SWTR	W	0h	0h: No software trigger request 1h: Sensor 2 Channel's software trigger request. The SWTR bit will be cleared automatically after the Sensor 2 software trigger request is in service. The S2_MTP_EN must active for SWTR to function

42.8.4.44 S3_MTP_EN Register (Offset = B00h) [Reset = 00000000h]

S3_MTP_EN is shown in Figure 42-76 and described in Table 42-79.

Return to the Summary Table.

Sensor 3 Channel MTP Enable Register

Figure 42-76. S3_MTP_EN Register

31	30	29	28	27	26	25	24
RSVD							
R-0h							
23	22	21	20	19	18	17	16
RSVD							
R-0h							
15	14	13	12	11	10	9	8
RSVD							
R-0h							
7	6	5	4	3	2	1	0
RSVD							MTP_EN
R-0h							R/W-0h

Table 42-79. S3_MTP_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RSVD	R	0h	Reserved bits
0	MTP_EN	R/W	0h	0h: No trigger pulses generated 1h: The Master Trigger Pulse Generator of the Sensor 3 Channel will generate trigger pulses when it is triggered via Hardware or Software

42.8.4.45 S3_MTP_CMP1 Register (Offset = B04h) [Reset = 00000000h]

S3_MTP_CMP1 is shown in Figure 42-77 and described in Table 42-80.

Return to the Summary Table.

Sensor 3 Channel MTP Compare1 Register

Figure 42-77. S3_MTP_CMP1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-80. S3_MTP_CMP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S3_MTP_CMP1 register contains the time (in clock tick unit) of the Sensor 3 Channel's Master Trigger Pulse Generator output to be toggled. No toggle if TOGGLETIME = 0.

42.8.4.46 S3_MTP_CMP2 Register (Offset = B08h) [Reset = 00000000h]

S3_MTP_CMP2 is shown in Figure 42-78 and described in Table 42-81.

Return to the Summary Table.

Sensor 3 Channel MTP Compare2 Register

Figure 42-78. S3_MTP_CMP2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-81. S3_MTP_CMP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S3_MTP_CMP2 register contains the time (in clock tick unit) of the Sensor 3 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S3_MTP_CMP1's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.47 S3_MTP_CMP3 Register (Offset = B0Ch) [Reset = 00000000h]

S3_MTP_CMP3 is shown in Figure 42-79 and described in Table 42-82.

Return to the Summary Table.

Sensor 3 Channel MTP Compare3 Register

Figure 42-79. S3_MTP_CMP3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-82. S3_MTP_CMP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S3_MTP_CMP3 register contains the time (in clock tick unit) of the Sensor 3 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S3_MTP_CMP2's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.48 S3_MTP_CMP4 Register (Offset = B10h) [Reset = 00000000h]

S3_MTP_CMP4 is shown in Figure 42-80 and described in Table 42-83.

Return to the Summary Table.

Sensor 3 Channel MTP Compare4 Register

Figure 42-80. S3_MTP_CMP4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-83. S3_MTP_CMP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S3_MTP_CMP4 register contains the time (in clock tick unit) of the Sensor 3 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S3_MTP_CMP3's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.49 S3_MTP_CMP5 Register (Offset = B14h) [Reset = 00000000h]

S3_MTP_CMP5 is shown in Figure 42-81 and described in Table 42-84.

Return to the Summary Table.

Sensor 3 Channel MTP Compare5 Register

Figure 42-81. S3_MTP_CMP5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-84. S3_MTP_CMP5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S3_MTP_CMP5 register contains the time (in clock tick unit) of the Sensor 3 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S3_MTP_CMP4's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.50 S3_MTP_CMP6 Register (Offset = B18h) [Reset = 00000000h]

S3_MTP_CMP6 is shown in Figure 42-82 and described in Table 42-85.

Return to the Summary Table.

Sensor 3 Channel MTP Compare6 Register

Figure 42-82. S3_MTP_CMP6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-85. S3_MTP_CMP6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S3_MTP_CMP6 register contains the time (in clock tick unit) of the Sensor 3 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S3_MTP_CMP5's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.51 S3_MTP_CMP7 Register (Offset = B1Ch) [Reset = 00000000h]

S3_MTP_CMP7 is shown in Figure 42-83 and described in Table 42-86.

Return to the Summary Table.

Sensor 3 Channel MTP Compare7 Register

Figure 42-83. S3_MTP_CMP7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-86. S3_MTP_CMP7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S3_MTP_CMP7 register contains the time (in clock tick unit) of the Sensor 3 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S3_MTP_CMP6's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.52 S3_MTP_CMP8 Register (Offset = B20h) [Reset = 00000000h]

S3_MTP_CMP8 is shown in Figure 42-84 and described in Table 42-87.

Return to the Summary Table.

Sensor 3 Channel MTP Compare8 Register

Figure 42-84. S3_MTP_CMP8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-87. S3_MTP_CMP8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S3_MTP_CMP8 register contains the time (in clock tick unit) of the Sensor 3 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S3_MTP_CMP7's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.53 S3_MTP_CMP9 Register (Offset = B24h) [Reset = 00000000h]

S3_MTP_CMP9 is shown in Figure 42-85 and described in Table 42-88.

Return to the Summary Table.

Sensor 3 Channel MTP Compare9 Register

Figure 42-85. S3_MTP_CMP9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-88. S3_MTP_CMP9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S3_MTP_CMP9 register contains the time (in clock tick unit) of the Sensor 3 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S3_MTP_CMP8's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.54 S3_MTP_CMP10RE Register (Offset = B28h) [Reset = 00000000h]

S3_MTP_CMP10RE is shown in Figure 42-86 and described in Table 42-89.

Return to the Summary Table.

Sensor 3 Channel MTP Compare10 Receiver Enable Register

Figure 42-86. S3_MTP_CMP10RE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-89. S3_MTP_CMP10RE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S3_MTP_CMP10RE defines when to enable the CSENT Receiver while the Sensor 3's trigger pulses are generated. This TOGGLETIME will be compared independently, since it might be higher or lower than other Sensor 3's compare register value. The value is specified in clock tick units.

42.8.4.55 S3_MTP_PERIOD Register (Offset = B2Ch) [Reset = 00000000h]

S3_MTP_PERIOD is shown in Figure 42-87 and described in Table 42-90.

Return to the Summary Table.

Sensor 3 Channel MTP Period Register

Figure 42-87. S3_MTP_PERIOD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					PERIOD										
R-0h																					R/W-0h										

Table 42-90. S3_MTP_PERIOD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	PERIOD	R/W	0h	PERIOD of the S3_MTP_PERIOD register indicates the total time that the master trigger pulse output waveform is active. The PERIOD must be larger than all Sensor 3 Channel Compares' TOGGLETIME value. The value is specified in clock tick units.

42.8.4.56 S3_MTP_TO Register (Offset = B30h) [Reset = 00000000h]

S3_MTP_TO is shown in Figure 42-88 and described in Table 42-91.

Return to the Summary Table.

Sensor 3 Channel MTP TimeOut Register

Figure 42-88. S3_MTP_TO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEOUT																															
R/W-0h																															

Table 42-91. S3_MTP_TO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TIMEOUT	R/W	0h	The TIMEOUT of the S3_MTP_TO register indicates a time to initiate an interrupt if the CSENT Receiver has not received response (low pulse of calibration/synchronization) from Sensor 3 within timeout time after master trigger pulse is generated. The value is specified in clock tick units.

42.8.4.57 S3_TRIGSEL Register (Offset = B34h) [Reset = 00000000h]

S3_TRIGSEL is shown in Figure 42-89 and described in Table 42-92.

Return to the Summary Table.

Sensor 3 Channel MTP Trigger Select Register

Figure 42-89. S3_TRIGSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																									TRIGSEL						
R-0h																									R/W-0h						

Table 42-92. S3_TRIGSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RSVD	R	0h	Reserved bits
6-0	TRIGSEL	R/W	0h	Specifies which hardware trigger source enables the Sensor 3 Channel Master Trigger Pulse Generator. The possible internal triggers are: 0h: Disable (no hardware triggers) 1h: Broadcast Trigger done 2h: Sensor 1 Trigger done 3h: Sensor 2 Trigger done 4h: Sensor 3 Trigger done 5h: Sensor 4 Trigger done External triggers are also available and are described by the 'Channel Triggers' section.

42.8.4.58 S3_MTP_SWTR Register (Offset = B38h) [Reset = 00000000h]

S3_MTP_SWTR is shown in Figure 42-90 and described in Table 42-93.

Return to the Summary Table.

Sensor 3 Channel MTP Software Trigger Register

Figure 42-90. S3_MTP_SWTR Register

31	30	29	28	27	26	25	24
RSVD							
W-0h							
23	22	21	20	19	18	17	16
RSVD							
W-0h							
15	14	13	12	11	10	9	8
RSVD							
W-0h							
7	6	5	4	3	2	1	0
RSVD							SWTR
W-0h							W-0h

Table 42-93. S3_MTP_SWTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RSVD	W	0h	Reserved bits
0	SWTR	W	0h	0h: No software trigger request 1h: Sensor 3 Channel's software trigger request. The SWTR bit will be cleared automatically after the Sensor 3 software trigger request is in service. The S3_MTP_EN must active for SWTR to function

42.8.4.59 S4_MTP_EN Register (Offset = C00h) [Reset = 00000000h]

S4_MTP_EN is shown in Figure 42-91 and described in Table 42-94.

Return to the Summary Table.

Sensor 4 Channel MTP Enable Register

Figure 42-91. S4_MTP_EN Register

31	30	29	28	27	26	25	24
RSVD							
R-0h							
23	22	21	20	19	18	17	16
RSVD							
R-0h							
15	14	13	12	11	10	9	8
RSVD							
R-0h							
7	6	5	4	3	2	1	0
RSVD							MTP_EN
R-0h							R/W-0h

Table 42-94. S4_MTP_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RSVD	R	0h	Reserved bits
0	MTP_EN	R/W	0h	0h: No trigger pulses generated 1h: The Master Trigger Pulse Generator of the Sensor 4 Channel will generate trigger pulses when it is triggered via Hardware or Software

42.8.4.60 S4_MTP_CMP1 Register (Offset = C04h) [Reset = 00000000h]

S4_MTP_CMP1 is shown in Figure 42-92 and described in Table 42-95.

Return to the Summary Table.

Sensor 4 Channel MTP Compare1 Register

Figure 42-92. S4_MTP_CMP1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-95. S4_MTP_CMP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S4_MTP_CMP1 register contains the time (in clock tick unit) of the Sensor 4 Channel's Master Trigger Pulse Generator output to be toggled. No toggle if TOGGLETIME = 0.

42.8.4.61 S4_MTP_CMP2 Register (Offset = C08h) [Reset = 00000000h]

S4_MTP_CMP2 is shown in Figure 42-93 and described in Table 42-96.

Return to the Summary Table.

Sensor 4 Channel MTP Compare2 Register

Figure 42-93. S4_MTP_CMP2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-96. S4_MTP_CMP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S4_MTP_CMP2 register contains the time (in clock tick unit) of the Sensor 4 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S4_MTP_CMP1's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.62 S4_MTP_CMP3 Register (Offset = C0Ch) [Reset = 00000000h]

S4_MTP_CMP3 is shown in Figure 42-94 and described in Table 42-97.

Return to the Summary Table.

Sensor 4 Channel MTP Compare3 Register

Figure 42-94. S4_MTP_CMP3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-97. S4_MTP_CMP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S4_MTP_CMP3 register contains the time (in clock tick unit) of the Sensor 4 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S4_MTP_CMP2's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.63 S4_MTP_CMP4 Register (Offset = C10h) [Reset = 00000000h]

S4_MTP_CMP4 is shown in Figure 42-95 and described in Table 42-98.

Return to the Summary Table.

Sensor 4 Channel MTP Compare4 Register

Figure 42-95. S4_MTP_CMP4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-98. S4_MTP_CMP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S4_MTP_CMP4 register contains the time (in clock tick unit) of the Sensor 4 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S4_MTP_CMP3's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.64 S4_MTP_CMP5 Register (Offset = C14h) [Reset = 00000000h]

S4_MTP_CMP5 is shown in Figure 42-96 and described in Table 42-99.

Return to the Summary Table.

Sensor 4 Channel MTP Compare5 Register

Figure 42-96. S4_MTP_CMP5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-99. S4_MTP_CMP5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S4_MTP_CMP5 register contains the time (in clock tick unit) of the Sensor 4 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S4_MTP_CMP4's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.65 S4_MTP_CMP6 Register (Offset = C18h) [Reset = 00000000h]

S4_MTP_CMP6 is shown in Figure 42-97 and described in Table 42-100.

Return to the Summary Table.

Sensor 4 Channel MTP Compare6 Register

Figure 42-97. S4_MTP_CMP6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-100. S4_MTP_CMP6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S4_MTP_CMP6 register contains the time (in clock tick unit) of the Sensor 4 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S4_MTP_CMP5's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.66 S4_MTP_CMP7 Register (Offset = C1Ch) [Reset = 00000000h]

S4_MTP_CMP7 is shown in Figure 42-98 and described in Table 42-101.

Return to the Summary Table.

Sensor 4 Channel MTP Compare7 Register

Figure 42-98. S4_MTP_CMP7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-101. S4_MTP_CMP7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S4_MTP_CMP7 register contains the time (in clock tick unit) of the Sensor 4 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S4_MTP_CMP6's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.67 S4_MTP_CMP8 Register (Offset = C20h) [Reset = 00000000h]

S4_MTP_CMP8 is shown in Figure 42-99 and described in Table 42-102.

Return to the Summary Table.

Sensor 4 Channel MTP Compare8 Register

Figure 42-99. S4_MTP_CMP8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-102. S4_MTP_CMP8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S4_MTP_CMP8 register contains the time (in clock tick unit) of the Sensor 4 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S4_MTP_CMP7's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.68 S4_MTP_CMP9 Register (Offset = C24h) [Reset = 00000000h]

S4_MTP_CMP9 is shown in Figure 42-100 and described in Table 42-103.

Return to the Summary Table.

Sensor 4 Channel MTP Compare9 Register

Figure 42-100. S4_MTP_CMP9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-103. S4_MTP_CMP9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S4_MTP_CMP9 register contains the time (in clock tick unit) of the Sensor 4 Channel's Master Trigger Pulse Generator output to be toggled. For non-zero TOGGLETIME, it has to be larger than S4_MTP_CMP8's TOGGLETIME. No toggle if TOGGLETIME = 0.

42.8.4.69 S4_MTP_CMP10RE Register (Offset = C28h) [Reset = 00000000h]

S4_MTP_CMP10RE is shown in Figure 42-101 and described in Table 42-104.

Return to the Summary Table.

Sensor 4 Channel MTP Compare10 Receiver Enable Register

Figure 42-101. S4_MTP_CMP10RE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					TOGGLETIME										
R-0h																					R/W-0h										

Table 42-104. S4_MTP_CMP10RE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	TOGGLETIME	R/W	0h	TOGGLETIME of the S4_MTP_CMP10RE defines when to enable the CSENT Receiver while the Sensor 4's trigger pulses are generated. This TOGGLETIME will be compared independently, since it might be higher or lower than other Sensor 4's compare register value. The value is specified in clock tick units.

42.8.4.70 S4_MTP_PERIOD Register (Offset = C2Ch) [Reset = 00000000h]

S4_MTP_PERIOD is shown in Figure 42-102 and described in Table 42-105.

Return to the Summary Table.

Sensor 4 Channel MTP Period Register

Figure 42-102. S4_MTP_PERIOD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					PERIOD										
R-0h																					R/W-0h										

Table 42-105. S4_MTP_PERIOD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD	R	0h	Reserved bits
10-0	PERIOD	R/W	0h	PERIOD of the S4_MTP_PERIOD register indicates the total time that the master trigger pulse output waveform is active. The PERIOD must be larger than all Sensor 4 Channel Compares' TOGGLETIME value. The value is specified in clock tick units.

42.8.4.71 S4_MTP_TO Register (Offset = C30h) [Reset = 00000000h]

S4_MTP_TO is shown in Figure 42-103 and described in Table 42-106.

Return to the Summary Table.

Sensor 4 Channel MTP TimeOut Register

Figure 42-103. S4_MTP_TO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEOUT																															
R/W-0h																															

Table 42-106. S4_MTP_TO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TIMEOUT	R/W	0h	The TIMEOUT of the S4_MTP_TO register indicates a time to initiate an interrupt if the CSENT Receiver has not received response (low pulse of calibration/synchronization) from Sensor 4 within timeout time after master trigger pulse is generated. The value is specified in clock tick units.

42.8.4.72 S4_TRIGSEL Register (Offset = C34h) [Reset = 00000000h]

S4_TRIGSEL is shown in Figure 42-104 and described in Table 42-107.

Return to the Summary Table.

Sensor 4 Channel MTP Trigger Select Register

Figure 42-104. S4_TRIGSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																									TRIGSEL						
R-0h																									R/W-0h						

Table 42-107. S4_TRIGSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RSVD	R	0h	Reserved bits
6-0	TRIGSEL	R/W	0h	Specifies which hardware trigger source enables the Sensor 4 Channel Master Trigger Pulse Generator. The possible internal triggers are: 0h: Disable (no hardware triggers) 1h: Broadcast Trigger done 2h: Sensor 1 Trigger done 3h: Sensor 2 Trigger done 4h: Sensor 3 Trigger done 5h: Sensor 4 Trigger done External triggers are also available and are described by the 'Channel Triggers' section.

42.8.4.73 S4_MTP_SWTR Register (Offset = C38h) [Reset = 00000000h]

S4_MTP_SWTR is shown in Figure 42-105 and described in Table 42-108.

Return to the Summary Table.

Sensor 4 Channel MTP Software Trigger Register

Figure 42-105. S4_MTP_SWTR Register

31	30	29	28	27	26	25	24
RSVD							
W-0h							
23	22	21	20	19	18	17	16
RSVD							
W-0h							
15	14	13	12	11	10	9	8
RSVD							
W-0h							
7	6	5	4	3	2	1	0
RSVD							SWTR
W-0h							W-0h

Table 42-108. S4_MTP_SWTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RSVD	W	0h	Reserved bits
0	SWTR	W	0h	0h: No software trigger request 1h: Sensor 4 Channel's software trigger request. The SWTR bit will be cleared automatically after the Sensor 4 software trigger request is in service. The S4_MTP_EN must active for SWTR to function

42.8.4.74 WAITTIME Register (Offset = F00h) [Reset = 00000100h]

WAITTIME is shown in Figure 42-106 and described in Table 42-109.

Return to the Summary Table.

Global Waittime Register

Figure 42-106. WAITTIME Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																WAITTIME															
R-0h																R/W-100h															

Table 42-109. WAITTIME Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RSVD	R	0h	Reserved bits
15-0	WAITTIME	R/W	100h	The Global WAITTIME contains the delay time between the last data frame received to the new master trigger pulse to be generated. The value is specified in clock tick units. After the Sensor's frame received, the trigger generator will wait for WAITTIME before processing the next trigger request.

42.8.4.75 TPGENSTAT Register (Offset = F04h) [Reset = 00000000h]

TPGENSTAT is shown in Figure 42-107 and described in Table 42-110.

Return to the Summary Table.

Master Trigger Pulse Global Status Register

Figure 42-107. TPGENSTAT Register

31	30	29	28	27	26	25	24
RSVD_1							
R-0h							
23	22	21	20	19	18	17	16
RSVD_1							
R-0h							
15	14	13	12	11	10	9	8
RSVD_1			TRIG_REQ				
R-0h			R-0h				
7	6	5	4	3	2	1	0
SWTSTAT	RSVD		ACTIVE_CH				
R-0h	R-0h		R-0h				

Table 42-110. TPGENSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RSVD_1	R	0h	Reserved bits
12-8	TRIG_REQ	R	0h	The TRIG_REQ tells which channel/channels receive trigger request (external /internal or software) that are pending to be serviced. 00000b: No trigger received 00001b: Broadcast channel receive trigger 00010b: Sensor 1 channel receive trigger 00100b: Sensor 2 channel receive trigger 01000b: Sensor 3 channel receive trigger 10000b: Sensor 4 channel receive trigger These TRIG_REQ bits will be 0s when there is no pending trigger request to be serviced.
7	SWTSTAT	R	0h	0h: No Software trigger 1h: The trigger is initiated via software (SWTR), and not through external/internal triggers. This bit will be cleared before the next trigger pulse is generated
6-5	RSVD	R	0h	Reserved bits
4-0	ACTIVE_CH	R	0h	ACTIVE_CH defines current channel trigger source that is in service (generating pulses). 00001b: Broadcast Channel 00010b: Sensor 1 Channel 00100b: Sensor 2 Channel 01000b: Sensor 3 Channel 10000b: Sensor 4 Channel

42.8.4.76 MTP_VERSION Register (Offset = F30h) [Reset = 01000001h]

MTP_VERSION is shown in Figure 42-108 and described in Table 42-111.

Return to the Summary Table.

MTPG IP Core Version Number Register

Figure 42-108. MTP_VERSION Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VERSION																															
R-01000001h																															

Table 42-111. MTP_VERSION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VERSION	R	01000001h	The VERSION can programatically provide the version number of the MTPG IP Core.

42.8.4.77 MTP_SWR Register (Offset = F3Ch) [Reset = 00000000h]

MTP_SWR is shown in Figure 42-109 and described in Table 42-112.

Return to the Summary Table.

MTPG Global Software Reset Register. Writing 1 to this register will cause Software Reset

Figure 42-109. MTP_SWR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD															SWR
W-0h															W-0h

Table 42-112. MTP_SWR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RSVD	W	0h	Reserved bits
0	SWR	W	0h	0h: No MTPG software reset 1h: It will cause MTPG software reset and all registers will have their default value

Chapter 43

► SECURITY PERIPHERALS

The following chapters describe the Security peripherals.

Technical Reference Manual Overview

The block diagram is shown in Figure 43-1. This Technical Reference Manual is organized into five major sections:

- C29x SYSTEM RESOURCES

  These chapters describe the C29x CPU subsystem, C29x Boot ROM, device configuration, and other system peripherals.

- ANALOG PERIPHERALS

  These chapters describe the general analog subsystem configuration, Analog-to-Digital Converter (ADC), Buffered Digital-to-Analog Converter (DAC), and Comparator Subsystem (CMPSS).

- CONTROL PERIPHERALS

  These chapters describe the Enhanced Capture (eCAP), High-Resolution Capture (HRCAP), Enhanced Pulse-Width Modulator (ePWM) with High-Resolution Pulse-Width Modulator (HRPWM), Enhanced Quadrature Encoder Pulse (eQEP), and Sigma Delta Filter Module (SDFM) peripherals.

- COMMUNICATION PERIPHERALS

  These chapters describe the communication peripherals available to the C29x subsystem such as the EtherCAT, FSI, I2C, PMBUS, UART, LIN, SPI, and SENT.

- SECURITY PERIPHERALS

  This chapter describes the safety peripherals available to the C29x subsystem such as the Hardware Security Module (HSM) and Cryptographic Accelerator.

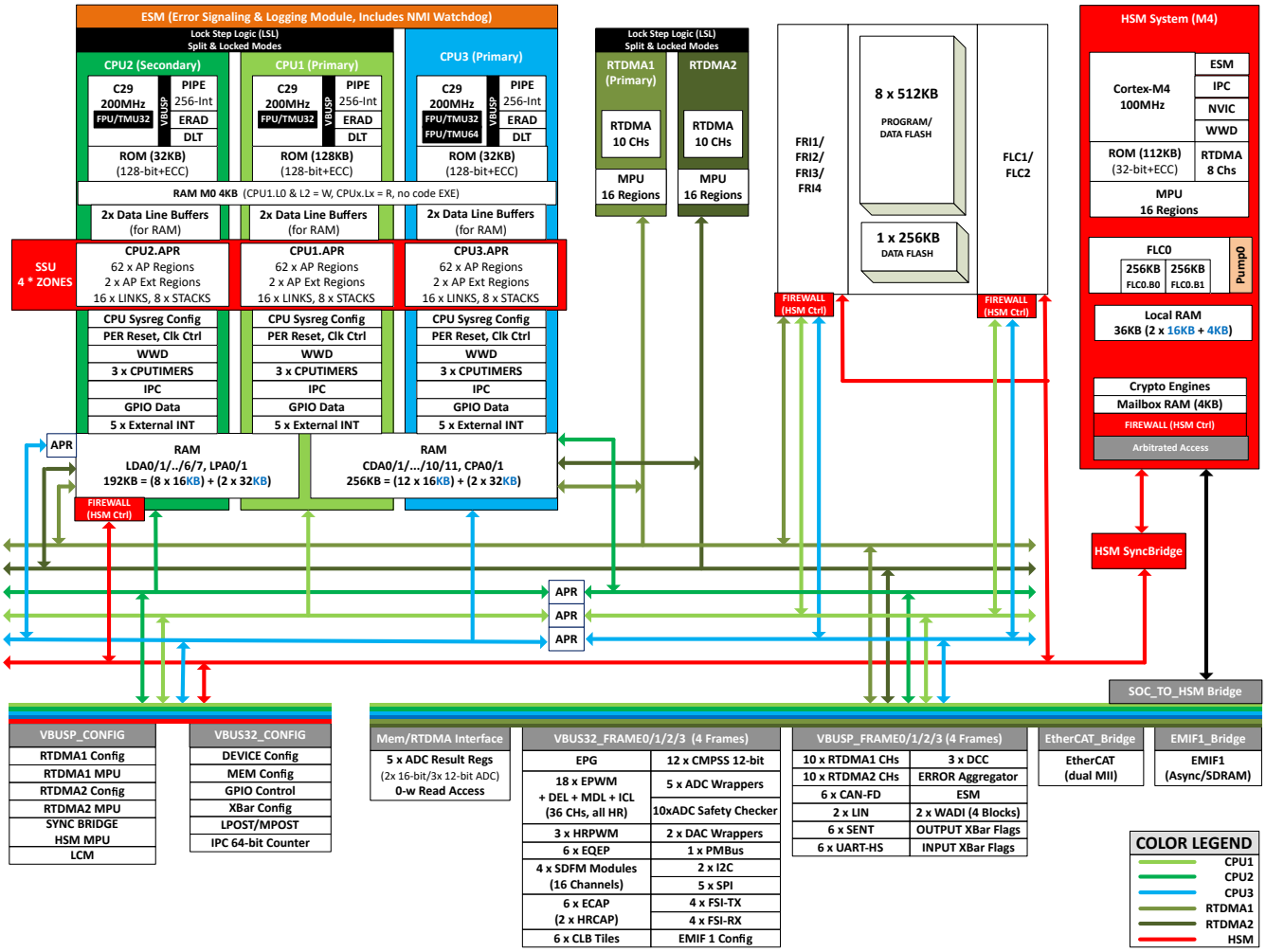


Figure 43-1. Block Diagram

Chapter 44

Security Modules

This chapter explains security modules for the C29x cores found on this MCU.

44.1 Hardware Security Module (HSM)

The Hardware Security Module (HSM) is a self-contained subsystem within the device that provides security and cryptographic functions. The host C29x subsystem interfaces with the HSM subsystem to perform the cryptographic operations required for code authentication, secure boot, secure firmware upgrades and encrypted run-time communications. A high-level view of the various subsystems in this device, with the HSM subsystem highlighted, is shown in Figure 44-1.

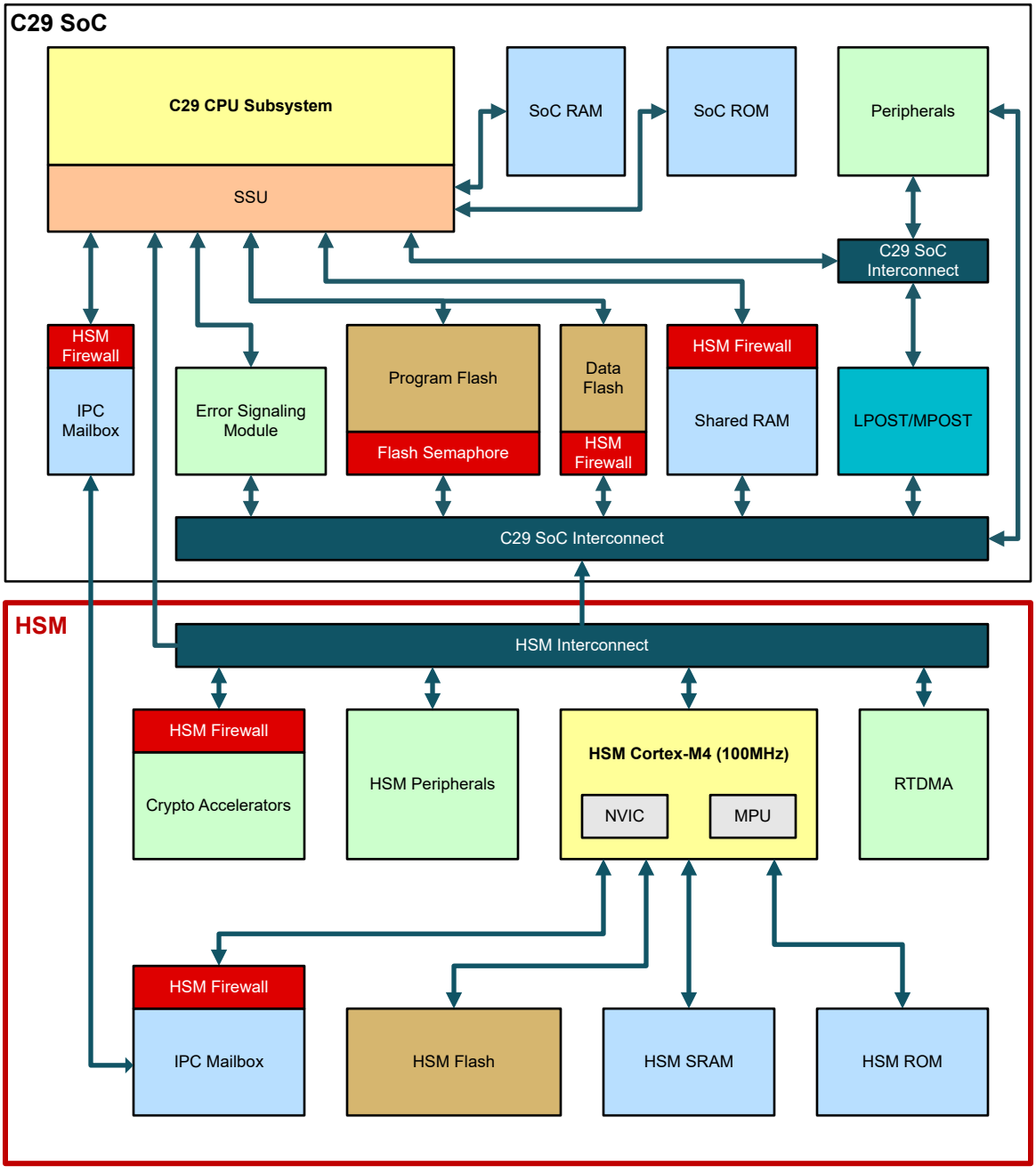


Figure 44-1. Device High-Level Block Diagram

At the center of the HSM is an ARM® Cortex®-M4 CPU running at 100MHz, with embedded SRAM, ROM, and up to 512KB of Flash memory. The Real-Time DMA (RTDMA) module enables fast data transfers between the HSM CPU and SRAM, HSM and application Flash memory banks, secure mailbox, and cryptographic engines.

The Security Manager module hosts the root-of-trust keys, defines the secure access mechanisms, controls the debug firewalls, and performs the security override sequences to establish protection of security assets if debug or failure-analysis operation is required.

The HSM includes a set of accelerator engines for executing cryptographic algorithms. These engines enable fast execution of symmetric encryption algorithms, hash functions, asymmetric encryption algorithms for public key infrastructure, and a true random number generator (TRNG). The Data Transform and Hashing Engine (DTHE) interfaces between the CPU and the cryptographic accelerators, providing interrupt and RTDMA trigger management and essential functions such as CRC and checksum computation.

In addition, the HSM provides peripheral modules to aid various security functions: timers, a real-time counter, a watchdog, DCC for clock monitoring, and ESM for error handling.

Communication between the HSM and the host application cores happens over a secure mailbox interface. The HSM controls various secure firewalls in the device, including the secure mailbox, cryptographic engines, shared RAM, and device Flash memory.

44.1.1 HSM Related Collateral

Foundational Materials

- C29x Academy - Hardware Security Module (HSM)

44.2 Cryptographic Accelerators

The Hardware Security Manager (HSM) includes several hardware accelerators to enable fast execution of key cryptographic algorithms. These engines are described in Table 44-1.

Table 44-1. List of Cryptographic Accelerator Engines

Engine	Algorithms Supported
AES (Advanced Encryption Standard)	Symmetric Algorithms: AES-128, AES-192, AES-256 Cipher modes: ECB, CTR, CBC, CFB, OFB, CCM, GCM Authentication: CBC-MAC
SM4	Symmetric Algorithms: SM4
PKE (Public Key Engine)	High-performance PKE for large-vector math/modulus operation Ciphers: RSA-2048, RSA-3092, RSA-4096, ECC (Curve25519, X25519, SecP256r1, secP256k1, secP384r1, secP384k1, Brain Pool, and more), SM2 Supports cryptographic operations: ECDSA, EdDSA, ECDH, EdDH, SM2DSA Side-channel protection (DPA, FIA)
SHA	Hash Algorithms: SHA-256, SHA-384, SHA-512 Keyed hashing: HMAC-SHA256, HMAC-SHA512
SM3	Hash Algorithms: SM3 (256 bits, 384 bits, 512 bits)
TRNG	True random number generator Deterministic random bit generator (DRBG)

Revision History

DATE	REVISION	NOTES
November 2024	*	Initial Release

This page intentionally left blank.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265